Apple Assembly Line

$1.80

Volume 7 -- Issue 4 January, 1987

In This Issue...

//gs I/O Soft Switches 2
Improved & Safer Text I/O Speedup for DOS 3.3. 8
Monitor CTRL-Y Command in the //gs 12
How to Use the USR Command in S-C Macro Assembler. 18
Little Bugs in ProDOS /RAM 22
New Features in Apple //gs Monitor 23
New ProDOS Version of S-C XREF 29

Jerry Lewis (not the entertainer), a subscriber from New Boston, Michigan, recently worked with Phil Wershba at SCRG to modify chip 0 on the QuikLoader card. The result is the ability to include ProDOS in chip 0. If you already have a QuikLoader, Phil says upgrades are available (there is a small fee this time). Call Phil at 1-800-635-8310. If you do not have a QuikLoader yet, and want one, you can order from us (we wil even save you a few dollars!).

Jim Davis (not the cartoonist), a subscriber in Upland, California, has a business that may be of interest to a lot of you. He will buy broken computers, especially Apple II and IIPlus models. He will also do chip-level repair of computers, power supplies, and 5.25" disk drives. If you have a II or IIPlus that will not quite work with a 65C02, and you wish it would, he will modify the motherboard for you so that it works. I haven't tried out his services yet, but if it sounds interesting call him at (714) 982-0860. Or write to him at Beehive Enterprises, 1476 N. Grove Ave., Upland, CA 91786.

//gs I/O Soft Switches.....Bob Sander-Cederlof

In spite of the ton of documentation Apple has made available to developers about the //gs, there is a lot of nitty-gritty detail that we need and have not got. Or at least it is very difficult to find the important, weed out the inaccurate, sift through the mutually contradictory, and so on.

I have been trying to get a complete and accurate map of the I/O addresses in the $C000-$C07F range, and I think I am getting close. Here is a summary of what I have learned.

The following addresses have exactly the same function in the //e and the //gs:

$C000 - $C01F	Keyboard & Memory mapping
$C030	Speaker Toggle
$C050 - $C05F	Graphics modes & Gameport ANx
$C061 - $C067	Gameport buttons & Paddles
$C070	Paddle Trigger
$C080 - $C08F	Language card control

The following //e addresses have no function in the //gs, but are reserved for future expansion:

$C020	was cassette output toggle
$C040	was gameport utility strobe

The address $C060 was used for cassette input in //e, II+, and II systems. In the //gs some sources say $C060 is used for a fourth gameport switch input, called switch 3 (I cannot find such a signal on any gameport pin in my //gs). $C061 through $C063 are still called switches 0, 1, and 2. $C061 is still attached to the Open-Apple key, and $C062 to the Solid-Apple.

$C071-$C07F is used by the //gs for ROM-based code, rather than soft swtiches. This range of memory is always available, so interrupts vector here.

```
C071: E2 40         SEP #$40    SET V-BIT
C073: 50            (BVC OPCODE, SKIPS NEXT BYTE)
C074: B8            CLV         CLR V-BIT
C075: 5C 10 00 E1   JMP $E10010
C079: 38            SEC
C07A: 90            (BCC OPCODE, SKIPS NEXT BYTE)
C07B: 18            CLC
C07C: 5C 5B B6 FF   JMP $FFB65B
```

Address $C021 controls the NTSC video output. Store $80 to kill color output, or $00 to enable color. This has no effect on the RGB output.

Address $C022 controls the RGB video output text and background color. The high four bits determine the text (foreground) color, and the low four bits determine the background color. The border color is controlled by the low four bits of $C034. In all of these, the color code is the same as lo-res graphics.

S-C Macro Assembler Version 2.0.....DOS $100, ProDOS $100, both for $120
Version 2.0 DOS Upgrade Kit for 1.0/1.1/1.2 owners.....$20
ProDOS Upgrade Kit for Version 2.0 DOS owners.....$30
Source Code of S-C Macro 2.0 (DOS only).....additional $100
Full Screen Editor for S-C Macro (with complete source code).....$49
S-C Cross Reference Utility.....without source code $20, with source $50
RAK-Ware DISASM.....without source code $30, with source $50
S-C Word Processor (with complete source code).....$50
DP18 Source and Object.....$50
Double Precision Floating Point for Applesoft (with source code).....$50
ES-CAPE (Extended S-C Applesoft Program Editor).....
Including Version 2.0 With Source Code.....$50
ES-CAPE Version 2.0 and Source Code Update (for Registered Owners)....$30
Bag of Tricks 2 (Quality Software).....($49.95) $45 *
MacASM -- Macro Assembler for MacIntosh (Mainstay).....($150.00) $100 *
S-C Documentor (complete commented source code of Applesoft ROMs).....$50
Cross Assemblers for owners of S-C Macro Assembler.....$32.50 to $50 each
(Available: 6800/1/2, 6301, 6805, 6809, 68000, Z-80, Z-8, 8048,
8051, 8085, 1802/4/5, PDP-11, GI1650/70, others)

AAL Quarterly Disks.....each $15, or any four for $45
Each disk contains the source code from three issues of AAL:
Jan-Mar, Apr-Jun, Jul-Sep, and Oct-Dec.
(All source code is formatted for S-C Macro Assembler. Other assemblers require some effort to convert file type and edit directives.)

Diskettes (with hub rings)..... package of 25 for $15 *
Vinyl disk pages, 6"x8.5", hold two disks each.....10 for $6 *
Diskette Mailing Protectors (hold 1 or 2 disks).....40 cents each
(Cardboard folders designed to fit 6"X9" Envelopes.) or $25 per 100 *
Envelopes for Diskette Mailers..... 6 cents each

65802 Microprocessor (Western Design Center).....($95) $50 *
quikLoader EPROM System (SCRG).....($179) $170 *
PROmGRAMER (SCRG).....($149.50) $140 *
Minuteman 250 Uninterruptible Power Supply.....($359) $320 +

"65816/65802 Assembly Language Programming", Fischer.....($19.95) $18 *
"Programming the 65816", Eyes.....($22.95) $21 *
"Apple //e Reference Manual", Apple Computer.....($24.95) $23 *
"Apple //c Reference Manual", Apple Computer.....($24.95) $23 *
"ProDOS Technical Reference Manual", Apple Computer.....($29.95) $27 *
"Apple ProDOS: Advanced Features for Programmers", Little..($17.95) $17 *
"ProDOS Inside and Out", Doms & Weishaar.....($16.95) $16 *
"DOStalk Scrapbook", Weishaar & Kersey.....($14.95) $14 *
"Beneath Apple ProDOS", Worth & Lechner.....($19.95) $18 *
"Beneath Apple DOS", Worth & Lechner.....($19.95) $18 *
"Inside the Apple //c", Little.....($19.95) $18 *
"Inside the Apple //e", Little.....($19.95) $18 *
"Understanding the Apple //e", Sather.....($24.95) $23 *
"Understanding the Apple II", Sather.....($22.95) $21 *
"Apple II+/IIe Troubleshooting & Repair Guide", Brenner.....($19.95) $18 *
"Apple][Circuit Description", Gayler.....($22.95) $21 *
"Assem. Language for Applesoft Programmers", Finley & Myers.($18.95) $18 *
"Now That You Know Apple Assembly Language...", Gilder.....($19.95) $18 *
"Enhancing Your Apple II, vol. 1", Lancaster.....($15.95) $15 *
"Enhancing Your Apple II, vol. 2", Lancaster.....($17.95) $17 *
"Assembly Cookbook for the Apple II/IIe", Lancaster.....($21.95) $20 *
"Real Time Programming -- Neglected Topics", Foster.....($9.95) $9 *
"Microcomputer Graphics", Myers.....($14.95) $14 *
"Assembly Lines -- the Book", Wagner.....($19.95) $18 *
"AppleVisions", Bishop & Grossberger.....($39.95) $36 *

* On these items add $2.00 for the first item and
$.75 for each additional item for US shipping.
\+ Inquire for shipping cost.
Foreign customers inquire for postage needed.
Texas residents please add 6 1/8 % sales tax to all orders.

*** S-C SOFTWARE, P. O. BOX 280300, Dallas, TX 75228 ***
*** (214) 324-2050 ***
*** Master Card, VISA, Discover and American Express ***

$C023 = VGC interrupt status and enable byte.

$C024 = Mouse data byte

$C025 = Keyboard data register
bit 7 = Open-Apple
bit 6 = Solid-Apple
bit 5 = flags this data updated
without a keypress
bit 4 = numeric keypad key depressed
bit 3 = repeat active
bit 2 = caps lock down
bit 1 = control key down
bit 0 = shift key down

$C026 = data register for talking to 50740 chip

$C027 = keyboard and mouse status byte

$C028 = ROM bank toggle. In the normal state with LCROM and I/O shadowing on addresses $D000-FFFF map directly into the same locations in bank $FF. After one LDA $C028, $D000-FFFF access $9000-BFFF in bank $FF. State of the toggle may be read in $C068 (state register) bit 1.

$C029 = control for new video states.
bit 7 = enable super hi-res
bit 6 = enable linear addressing for hi-res
bit 5 = force monochrome in super hi-res
bit 0 = enable bank 1 latch
Whatever you do to the data in this byte, be sure you do not change bit 0. First read the byte, then change bits 5,6,7 as you like, then store the byte.

$C02A = reserved

$C02B = foreign language selection
bits 765 = 0 for USA English
= 1 for UK English
= 2 for French
= 3 for Danish
= 4 for Spanish
= 5 for Italian
= 6 for German
= 7 for Swedish
bit 4 = 0 for NTSC, = 1 for PAL
bit 3 = 1 for above languages, = 0 for USA
The ASCII codes which change are $23, 40, 5B, 5C, 5D, 60, 7B, 7C, 7D, and 7E.

$C02C = Address for test mode read of character ROM

$C02D = slot ROM enables, where bit x controls slot x.
(do not enable slot 0 or 3!)

$C02E = vertical video address / 2

```
$C02F = horizontal video address in bits 6-0
       lsb of vertical address in bit 7
       In NTSC mode, the vertical address runs from $0FA
       through $1FF, for 262 lines.  The horizontal address
       runs from 40 through 7F, and 00 (65 different values)

$C031 = 3.5" disk drive control
       bit 7 = head select
       bit 6 = drive enable

$C032 = Clear Scan interrupt
       bit 6 = 0 to clear 1 second interrupt
       bit 5 = 0 to clear scanline interrupt

$C033 = clock chip data register

$C034 = clock chip control register & border color
       bit 7 = 1 to strobe data into clock
             = 0 when data has been accepted
       bit 6 = 1 to read chip, =0 to write chip
       bit 5 = chip enable after transfer
       bits 3-0 = border color

$C035 = shadow control register
       bit 6 = 0 to shadow I/O and "Language card".  When IOLC
                 shadowing is on, the RAM at $C000-CFFF is
                 folded into a second bank at $D000-DFFF,
                 making a "language card".
       bit 4 = 0 to shadow aux hi-res page (used with bits 1
                 and 2 to inhibit shadowing in bank $01)
       bit 3 = 0 to shadow 32K video buffer ($2000-9FFF in
                 bank $00)
       bit 2 = 0 to shadow hi-res page 2 ($4000-5FF in banks
                 $00 and $01)
       bit 1 = 0 to shadow hi-res page 1 ($2000-3FFF in banks
                 $00 and $01)
       bit 0 = 0 to shadow text pages ($400-7FF in both banks
                 $00 and $01.  Text page 2, $800-BFF, is never
                 shadowed.)
       RESET clears this register, enabling all shadowing.

$C036 = speed control and disk motor on detection
       bit 7 = 0 for slow speed
       bit 4 = 0 (if you make it =1, it will stop you cold!  I
                 got FATAL SYSTEM ERROR 0681.  Supposedly it
                 causes shadowing in ALL banks!)
       bits 3-0 = motor on detection bits for slots 7-4.  For
                 example, if bit 2 is on, the system will slow
                 down to 1MHz operation when the address $C0E9
                 (slot 6 motor-on command) is detected.  The
                 system will speed up again when $C0E8 (slot 6
                 motor-off) is detected, unless bit 7 = 0.

$C037 = DMA bank address register

$C038 = SCC channel B command register
$C039 = SCC channel A command register
$C03A = SCC channel B data register
```

7 Meg IIe/1 Meg IIc

Want the best & easiest-to-use (most support) RAM card on the market — at the best price? Buy Checkmate Technology's State-of-the-Art MULTIRAM RGB RAM CARD™ from us w/BATTERY BACKED-UP RAM options that save programs like AppleWorks for 10+ years! FAST CHEAP REPLACEMENT FOR HARD DISKS, USER EXPANDABLE TO 7 MEGABYTES, compatible with all (100%) 3rd party software/hardware, optional 65C816 Co-Processor card ($147), sharp 80 columns/Double Hi-Res, & FREE RGB*! It's a direct substitute for Ramworks I-III™ by Applied Engineering (also known as Preferred Computing) or Apple 80 col cards, has a **5 YR WARRANTY, 15 day money back guarantee, & FREE AUTOMATIC SOFTWARE UPDATES ONLY FROM US!**

MultiRam RGB expands to 1 Meg RAM + optional 4 Meg's **MultiRam Plus BATTERY PROTECTED RAM.** Optional 2 Meg's STATIC RAM here soon — for a total of 7 megabytes! **MultiRam IIe** expands to 768k. And the **software & manual is the best on the market!**

FREE APPLEWORKS EXPANDER SOFTWARE WITH EVERY CARD — loads ALL versions of Appleworks (in 30 seconds w/AW 2.0), even printer routines, runs faster, expands Desktop over 2048k, stores 23,000+ records & 4,800+ word processor lines, auto-segments large files onto multiple disks! **FREE** AW TIME/DAY/DATE ON-SCREEN w/ clock, AUTOCOPY TO RAM, PRODOS/DOS 3.3. RAM DISK & RAM TEST. Optional CP/M & Pascal Ram disks ($20/ea)!

	MultiRam RGB Card	MultiRam IIe Card
64k **MultiRam**	169.	129.
320k **MultiRam**	199.	164.
576k **MultiRam**	229.	199.
832k **MultiRam**	254.	234.
1024k MultiRam	**279.** ⬅ ⬅ ⬅ ⬅	

	MultiRam Plus Piggyback	
256k **MultiRam Plus**	220.	***NEW***
512k **MultiRam Plus**	269.	***NEW***
768k **MultiRam Plus**	316.	***NEW***
1024k **MultiRam Plus**	359.	***NEW***
2-4 meg **MultiRam Plus**	Call	***NEW***
Battery/AC Kit (for Plus)	36.	***NEW***

Apple IIe Enhancement Kit	**62.**
Cermetek 1200 Baud internal modem IIe/II+/IIgs	**179.**
Incomm 2400 Baud external univ. modem	**319.**
Clockworks Clock (Thunder/Time HO™ compat)	**89.**
13" RGB Monitor (free cable)	**322.**
ProAPP 10/20 Meg Hard Disks (+ FREE software)	**Call**
Apple Compatible Drives IIe/II+ (IIgs/IIc +15)	**135.**
NETWORKING UP TO 256 APPLE IIc/IIe/IIgs	**Call**

Terms: Add $4-Ground or $6-Air shipping & phone # to each U.S. MultiRam card order (foreign orders/FPO/APO extra). Add 3% for MasterCard/Visa (include #/expir) & P.O.'s. For fast delivery send Cashier's/Certified check/Money Order. C.O.D. (add $6) & personal checks accepted (allow 16 days). Tex res add 7 ¼% tax.

Checkmate Technology's MultiRam CX card easily expands your IIc to 640k and has a **CX + Piggyback** to add another 512k **(over 1152k total)!** It's 100% compatible with all IIc software/hardware & upgradable with a 65C816 kit ($105) — all using about 50% less power than Z-RAM! It has the **SAME FREE SOFTWARE/UPDATES & 5 YR WARRANTY AS IIe CARDS.**

256k MULTIRAM CX	**179.**
512k MULTIRAM CX	**209.**
768k MULTIRAM CX Combo	**328.**
1024k MULTIRAM CX Combo	**354.**
256k MULTIRAM CX+	**210.**
512k MULTIRAM CX+	**222.**
AutoWorks (Best AppleWorks Macro+)	**37.**
1200 Baud IIc Modem (+$20/cable)	**159.**
Clip-on IIc Rechargable Battery	**124.**
IIc Flat Panel Display (C-Vue)	**299.**
IIc System Clock	**66.**

WHY BUY FROM COIT VALLEY COMPUTERS RATHER THAN SOME MAIL ORDER HOUSES? Only we offer a **15 day MultiRam complete money back satisfaction guarantee,** extra software, more support, **free automatic software updates,** free 64k w/ each 256k/512k/768k IIe/RGB card. We know the products, & stock them for fast shipping! CALL FOR QUESTIONS, PRICES, DISCOUNTS. **SCHOOLS & GROUPS WELCOME.**

Autoworks, Ramworks III/Timemaster II H.O./Z-ram, respective trademarks of Software Touch, Applied Engineering. *Call for RGB Connectors/cables (not included).

ORDER NOW!

COIT VALLEY COMPUTERS (214) 234-5047
14055 Waterfall Way Dallas, Texas 75240

NAME _____

ADDRESS _____

CITY _____ STATE _____ ZIP _____

PHONE ( ) _____

SIGNATURE

QTY	DESCRIPTION	PRICE
MC/VS _____	SHIP.	
_____ EXP _____	TOTAL	

COIT VALLEY COMPUTERS
(214) 234-5047 MCI 2969684 Telex 6502969684 (WUI)
14055 Waterfall Way
Dallas, Texas 75240

```
$C03B = SCC channel A data register

$C03C = sound control
        bit 7 = 1 if busy
        bit 6 = 1 to enable auto-increment
        bit 5 = 0 to access chip, =1 to access RAM
        bits 3-0 = volume of DAC

$C03D = sound data register
$C03E = sound address low
$C03F = sound address high

$C041-$C048 used for mouse
$C049-$C04F reserved

$C068 = state register (both read and write)
                       same as
                       bit 7 of
        bit 7 = ALTZP   $C016
        bit 6 = PAGE2   $C01C
        bit 5 = RAMRD   $C013
        bit 4 = RAMWRT  $C014
        bit 3 = RDROM   $C012
        bit 2 = LCBANK  $C011
        bit 1 = ROMBNK  ----- reads $C028 toggle
        bit 0 = SLOTCX  $C015

$C069-$C06C reserved

$C06D = test mode bit register
$C06E = clear test mode
$C06F = set test mode
```

I don't have the foggiest idea what the test mode is or how to use it, but it sounds interesting.

I hope with this information as a start, you can learn a lot more about your //gs. If you find any errors above, please let us know.

	C02x	C03x	C04x	C05x	C06x
0	--rsvd--	Speaker	--rsvd--	Graphics	Switch 3??
1	MONO	3.5 Disk	Mouse Int	Text	Switch 1
2	TXT Color	Scan Int	--rsvd--	Unmixed	Switch 1
3	VGC Int	Clock Data	--rsvd--	Mixed	Switch 2
4	MouseData	Clock Ctrl	M Delta X	Text 1	Paddle 0
5	Key Mods	Shadow	M Delta Y	Text 2	Paddle 1
6	KM Data	Speed	M Flags	Lo Res	Paddle 2
7	KM Status	DMA Bank	VBL Int	Hi Res	Paddle 3
8	ROM Bank	SCC B Cmd	XY Int	Clr AN0	State Reg
9	New Video	SCC A Cmd	--rsvd--	Set AN0	--rsvd--
A	--rsvd--	SCC B Data	--rsvd--	Clr AN1	--rsvd--
B	Lang Sel	SCC A Data	--rsvd--	Set AN1	--rsvd--
C	CharROM	Sound Ctrl	--rsvd--	Clr AN2	--rsvd--
D	Slot ROM	Sound Data	--rsvd--	Set AN2	Test Mode
E	Vert Cnt	Sound AdrL	--rsvd--	Clr AN3	Test Off
F	Horiz Cnt	Sound AdrH	--rsvd--	Set AN3	Test On

Improved & Safer Text I/O Speedup for DOS.....Louis Pitz

I have been trying to piece together a lot of DOS 3.3 improvements published in back issues of Apple Assembly Line, and I especially wanted to fit the Text File I/O improvements made by Paul Schlyter. (See AAL July 1983: "Speeding Up Text File I/O", by Paul Schlyter; and "Making Paul's Patches Fit in DOS", by Bob Sander-Cederlof.) Bob has said in several phone conversations that he continues to have reservations about using this patch, because of the possibility of DOS becoming confused over the state of the File Manager Workarea. He even suggesting using the space for some other improvements.

Why didn't I follow his advice? I guess the idea of a faster way of handling TEXT files is so attractive that I keep thinking the approach can be made to work. I made some small changes to the 1983 patches, and I believe they are now safe. Bob's last-minute warning on page 17 of the 1983 article still holds, however: do not try to install the patch from inside an EXEC file. You should not patch a piece of code while it is being used! The code as it is here can be BRUN from within a small HELLO program with no trouble.

The boxed comment on page 16 of the 1983 article, about patching $A1B1 to $14, was interesting. This value determines how many locations will be cleared to zero by the loop starting at $A1AE. The loop as coded in DOS clears too many bytes, so Bob's patch fixes it. Then he can use those bytes at the end for a variable of his own. Another loop at $ABDC also clears out the location Bob wanted to use, but it did not interfere with Bob's use. I chose to avoid the problem by using a different location for my PNTR.SAVE variable: $B5FE,B5FF. This is still close enough to $B5C7 (start of FM workarea) to use the indexing trick (at lines 1740,1750 in my program).

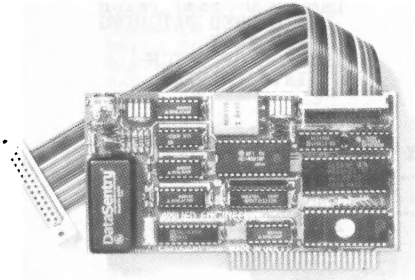
I saved the one extra byte Bob mentioned on page 15 of his article, by avoiding the SEC in the CHECK.OPCODE subroutine. I changed the origin of the patch from $B6B3 to $B6B4, to give one more byte to another set of patches I am using.

Bob's main reservation about using the speedup was the danger of FLAG being set without a valid address in PNTR.SAVE. I avoid this by not setting FLAG until after placing a valid address in the pointer (lines 1820-1840), and by clearing FLAG before using the pointer to save the workarea (lines 1730-1760).

FLAG is cleared by all FM commands other than Read and Write, including INIT, so there is no way an initialized disk will have the sign bit of FLAG set. If you follow all abnormal exits from handling text files with the CLOSE command, that will clear FLAG in those cases; it was good policy to do this even before these patches. [Maybe we should find all those abnormal exits, and make them specifically clear FLAG...]

I hope you like my version and agree that the enhancement can be safely used. All my tests, so far, indicate it is safe.

Serial Pro®
The intelligent multifunction card from Applied Engineering.



Serial Pro is a powerful multifunction card for the Apple IIe, II+, IIGs and compatible computers. Serial Pro combines a powerful serial port for interfacing to a printer, modem or other serial device, with a full function clock/calendar. Serial Pro's serial port is compatible with virtually all letter quality and dot matrix printers, including Apple's Imagewriter and Imagewriter II, and can communicate with the fastest external modems at baud rates from 50 to 19,200.

Set up is easy, because unlike other cards, there are no switches to program serial parameters or disks to use. Serial Pro has a built-in setup screen that allows all parameters to be set. And on-screen help that makes it super-easy. An on-board battery stores your selections for over 20 years, but you can change them at any time. Serial Pro is the perfect upgrade for Super Serial card owners looking for more performance or a built-in clock.

Serial Port Features • Fully Super Serial Card compatible but more versatile and easier to use • Compatible with printers • Compatible with modems • Prints graphics to printer (without software) in normal mode, 90° rotation, double size, side by side, and black/white inversion • Variable word length, number of stop bits and parity selection • Crystal controlled circuits are used to provide ultra accurate baud rates.

Clock Calendar Features • ProDos, DOS and AppleWorks compatible. Displays time and date on the AppleWorks screen • Automatically time and date stamps AppleWorks files • 20 year auto recharging nickel-cadmium battery • Realtime interrupts • Built-in on screen time and date settings (no software needed) • Does not use phantom slots.

Serial Pro is perfect for Apple IIe, II+, IIGs or compatible owners in need of a serial port or clock/calendar or both. But unlike other multifunction cards on the market, Serial Pro does not use phantom slots, so all the slots in your computer are useable.

Serial Pro comes complete with manual, cable and instructions for connecting to all the leading printers and modems.

Serial Pro **$159**

Order Today! See your dealer or call Applied Engineering, 9 a.m. to 11 p.m. 7 days. Or send check or money order to Applied Engineering, MasterCard, VISA and C.O.D. welcome. Texas residents add 5⅛% sales tax. Add $10.00 outside U.S.A.

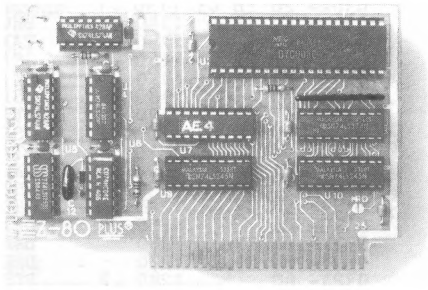
APPLIED ENGINEERING™

The Apple enhancement experts.

(214) 241-6060

P.O. Box 798, Carrollton, TX 75006

With the Z-80 Plus®, run over 5000 new CP/M® programs.



Now, get two computers in one, and all the advantages of both.

Enter the CP/M world with the new Z-80 Plus card from Applied Engineering, and introduce your Apple IIe, IIGs or II+ to thousands of CP/M programs. Only the Z-80 Plus comes standard with the new 5.1 software, the most advanced system ever for running CP/M programs.

The new 5.1 boasts advanced features like built-in disk emulation for popular memory expansion boards, boosting both system speed and storage capacity. And menu-driven utilities that let you get to work faster. The Z-80 Plus also lets you run older CP/M programs—all the way down to Version 1.6 (2.2 is the most popular).

The Z-80 Plus is the only card on the market capable of accessing more than 64K. If you have an extended 80-column card, all 128K is usable. And if you have RamWorks, RamFactor, GSRam, GSRam Plus, or an Apple memory card in your IIe, IIGs or II+, up to 1088K is available.

Each Z-80 Plus comes with our CP/M Ram Drive software, enabling most memory cards to work as a high-speed Ram disk so CP/M software can run up to *twenty times faster.* So packages like WordStar, dBASE II, Turbo Pascal and most others run at blinding speed.

Simply plug the Z-80 Plus into any slot in your Apple. You'll get the benefits of two computers in one—all at an unbelievably low price (only $149!).

• Fully compatible with ALL CP/M software • Fully compatible with 5¼" *and* 3½" disk drives as well as most hard disks including the Sider • Fully compatible with Microsoft disks (no pre-boot required) • Specifically designed for high speed operation • Runs WordStar, dBASE II, Turbo Pascal, and ALL other CP/M software with *no pre-boot.* • Semi-custom I.C. and low parts count allows Z-80 Plus to fly through CP/M programs with extremely low power consumption (we use the Z-80B) • Does EVERYTHING other Z-80 boards do, *plus* Z-80 interrupts • Five-year warranty

To get your Z-80 Plus or for more information see your dealer, or call 214-241-6060 9 a.m. to 11 p.m. seven days, or send check or money order to Applied Engineering. MasterCard, VISA and C.O.D. welcome. Texas residents add 5⅛% sales tax. Add $10.00 outside U.S.A.

Applied Engineering™

P.O. Box 798, Carrollton, TX 75006

(214) 241-6060

CPM is a registered trademark of Digital Research.

```
                        1000 *SAVE S.FAST TEXT (PITZ)
                        1010 *-------------------------------
                        1020         .OR $300
                        1030 *-------------------------------
00-                     1040 PPNTR   .EQ $00,01
02-                     1050 PATCH   .EQ $02,03
                        1060 *-------------------------------
                        1070 PATCHER
0300- A9 33             1080         LDA #PATCHES-1    POINT TO SET OF PATCHES
0302- 85 00             1090         STA PPNTR
0304- A9 03             1100         LDA /PATCHES-1
0306- 85 01             1110         STA PPNTR+1
0308- A0 00             1120         LDY #0            Y = 0 FROM NOW ON
030A- 20 2B 03          1130 .1      JSR GET.BYTE      LENGTH OF NEXT PATCH
030D- F0 1B             1140         BEQ .4            ...FINISHED PATCHING
030F- AA                1150         TAX               LENGTH IN X
0310- 20 2B 03          1160         JSR GET.BYTE      ADDRESS OF PATCH
0313- 85 02             1170         STA PATCH         ...INTO POINTER
0315- 20 2B 03          1180         JSR GET.BYTE
0318- 85 03             1190         STA PATCH+1
031A- 20 2B 03          1200 .2      JSR GET.BYTE      NEXT BYTE OF PATCH
031D- 91 02             1210         STA (PATCH),Y     TUCK IT AWAY
031F- E6 02             1220         INC PATCH         POINT TO NEXT ONE
0321- D0 02             1230         BNE .3
0323- E6 03             1240         INC PATCH+1
0325- CA                1250 .3      DEX               ARE WE FINISHED?
0326- D0 F2             1260         BNE .2            ...NO
0328- F0 E0             1270         BEQ .1           ...ALWAYS
032A- 60                1280 .4      RTS
                        1290 *-------------------------------
                        1300 GET.BYTE
032B- E6 00             1310         INC PPNTR         ADVANCE POINTER
032D- D0 02             1320         BNE .1
032F- E6 01             1330         INC PPNTR+1
0331- B1 00             1340 .1      LDA (PPNTR),Y     GET BYTE
0333- 60                1350         RTS
                        1360 *-------------------------------
                        1370         .MA PATCH
                        1380         .DA #LEN.]1,]2
                        1390 PAT.]1  .PH ]2
                        1400         .EM
                        1410 *-------------------------------
                        1420         .MA ENDP
                        1430         .EP
                        1440 LEN.]1  .EQ *-PAT.]1
                        1450         .EM
                        1460 *-------------------------------
                        1470 PATCHES
                        1480 *-------------------------------
0334-                   1490         >PATCH 1,$B6B4
0334- 49 B4 B6          0000>        .DA #LEN.1,$B6B4
                        0000> PAT.1  .PH $B6B4
                        1500 *-------------------------------
                        1510 *    Paul Schlyter's Text File Speed-up,
                        1520 *    as modified by Bob Sander-Cederlof (June 1983)
                        1530 *    as modified by Louis Pitz      (December 1986)
                        1540 *-------------------------------
42-                     1550 PNTR    .EQ $42,43
                        1560 *-------------------------------
AE6A-                   1570 COPY.BUFFER.TO.WORKAREA .EQ $AE6A
AE81-                   1580 SAVE.WORKAREA           .EQ $AE81
AF08-                   1590 POINT.TO.WORKAREA       .EQ $AF08
AF12-                   1600 SETUP.PNTR              .EQ $AF12
B5BB-                   1610 FM.OPCODE               .EQ $B5BB
B5C7-                   1620 FM.WORKAREA             .EQ $B5C7
B5FE-                   1630 PNTR.SAVE               .EQ $B5FE,B5FF    ****
                        1640 *-------------------------------
B6B4- 20 E8 B6          1650 PATCH1 JSR CHECK.OPCODE
B6B7- 90 0A             1660         BCC .1       NOT READ/WRITE
B6B9- CD FE B5          1670         CMP PNTR.SAVE
B6BC- D0 05             1680         BNE .1       NO
B6BE- EC FF B5          1690         CPX PNTR.SAVE+1
B6C1- F0 24             1700         BEQ PATCH3   YES, RETURN NOW
B6C3- 2C FC B6          1710 .1      BIT FLAG
B6C6- 10 0B             1720         BPL .2
B6C8- 4E FC B6          1730         LSR FLAG     CLEAR SIGN BIT     ****
```

```
B6CB- A2 37     1740         LDX #PNTR.SAVE-FM.WORKAREA
B6CD- 20 12 AF  1750         JSR SETUP.PNTR
B6D0- 20 81 AE  1760         JSR SAVE.WORKAREA                  ****
B6D3- 4C 6A AE  1770 .2      JMP COPY.BUFFER.TO.WORKAREA
                1780 *-------------------------------
B6D6- 20 E8 B6  1790 PATCH2  JSR CHECK.OPCODE
B6D9- B0 03     1800         BCS .1          ... READ OR WRITE  ****
B6DB- 4C 81 AE  1810         JMP SAVE.WORKAREA                  ****
B6DE- 8D FE B5  1820 .1      STA PNTR.SAVE
B6E1- 8E FF B5  1830         STX PNTR.SAVE+1
B6E4- 6E FC B6  1840         ROR FLAG        SET SIGN BIT
B6E7- 60        1850 PATCH3  RTS
                1860 *-------------------------------
                1870 CHECK.OPCODE
B6E8- 20 08 AF  1880         JSR POINT.TO.WORKAREA
B6EB- AD BB B5  1890         LDA FM.OPCODE
B6EE- C9 03     1900         CMP #3          READ?              ****
B6F0- F0 05     1910         BEQ .1          YES
B6F2- C9 04     1920         CMP #4          WRITE?             ****
B6F4- F0 01     1930         BEQ .1
B6F6- 18        1940         CLC
B6F7- A5 42     1950 .1      LDA PNTR
B6F9- A6 43     1960         LDX PNTR+1
B6FB- 60        1970         RTS
                1980 *-------------------------------
B6FC- 00        1990 FLAG    .HS 00          MUST START WITH FLAG=0
                2000 *-------------------------------
B6FD-           2010         >ENDP 1
                0000>        .EP
49-             0000> LEN.1  .EQ *-PAT.1
                2020 *-------------------------------

0380-           2040         >PATCH 2,$AB0B
AB0B- B4 B6     2050         .DA PATCH1
AB0D-           2060         >ENDP 2
                2070 *-------------------------------
0385-           2080         >PATCH 3,$B38F
B38F- D6 B6     2090         .DA PATCH2
B391-           2100         >ENDP 3
                2110 *-------------------------------
038A- 00        2120         .DA #0          <<< END OF PATCHES >>>
                2130 *-------------------------------
```


RAMWORKS™

ACCEPT NO SUBSTITUTES. BECAUSE THERE AREN'T ANY.

There's only one card like RamWorks. We've got the best hardware design. We supply the best software and we've got the best support from software companies.

If someone tempts you with an imitation, please get both sides of the story. You'll discover why RamWorks offers the best enhancements to AppleWorks and other programs, and at the lowest price.

GUARANTEED!

214-241-6060
9 AM - 11 PM

"We Set the Standard"
APPLIED ENGINEERING

The Monitor ctrl-Y Command in the //gs....Bob Sander-Cederlof

The Apple monitor has always had a control-Y command, called the "user" command. It allows you to add at least one command to the monitor on a temporary basis. The older versions merely JMPed to $03F8, but the //gs does a little more.

If you ran my program from last month's AAL, which printed out the addresses in the //gs monitor command table for each command, you found out that the ctrl-Y command starts at $FE13. To see what that is, enter the monitor L command as follows:

```
*FF/FE13L
```

The code in my //gs is CLC, XCE, and JMP $E100A0. The instruction installed at $E100A0 when you turn on the //gs is JMP $FFFE19, and the code at FFFE19 is SEC, XCE, and JMP $03F8. So we end up at $3F8 after all!

If you want to, you can ignore the new stuff and just use ctrl-Y like you did in the old days. Or, you can put a 3-byte address for a Native mode program into the JMP instruction at $E100A0.

I wrote a program to link into the $E100A0 vector and print out the registers, the mode, and the M and X status bits. I wanted to see for certain in what state the monitor called my ctrl-Y code.

Lines 1080-1150 install the address of my ctrl-Y code in the bank $E1 vector.

Lines 1180-1240 save the mode and the three main registers. Lines 1180-1210 on entry operate together with lines 1590-1610 at the exit, allowing me to run in Emulation mode regardless of the mode my caller was in. My program returns to the caller's mode and mx-state when it is finished.

Lines 1250-1350 show the A-, X-, and Y-registers as they were upon entry. I am only showing the low-byte of each of these, even though in the 65816 they have two bytes. Shame on me!

Lines 1410-1540 print either "EMULATION MODE" or "NATIVE MODE: M=1, X=1". If you write a little driver program to call MY.Y with various other combinations of M and X, you will see that MY.Y prints them as you have them.

Lines 1640-1680 print a message followed by the contents of the A-register in hexadecimal. I use it to print the contents of the three registers.

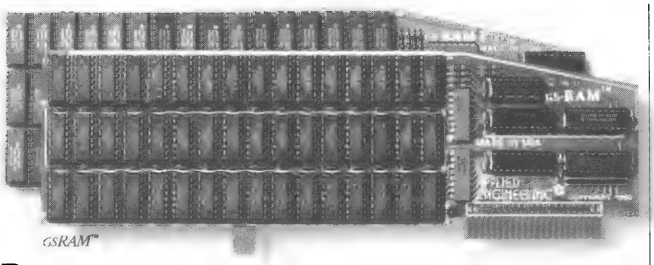
Lines 1700-1740 print "0" if the Z status bit is set upon entry; otherwise, they print a "1".

Lines 1760-1800 are just about the shortest loop for printing out an ASCII string I have come up with in my ten years of Apple programming. Strings are assembled with bit 7 high, and terminated by a 00 byte. On entry, the Y-register points to

Insist on GSRAM™ When You Buy Your IIGS™

Expand the IIGS RAM and ROM with the GSRAM or GSRAM Plus with ROM Pak. Available now with 256K to 8 MEG!

GSRAM Plus™



Remember the 16K cards for the II+ and the 64K cards for the IIe? At the time, that much memory seemed like a lot. But when the owners of these memory cards came to us for more memory, many had to throw away their smaller Apple memory cards or try to sell them. Most of our customers told us that had they known about Applied Engineering's larger memory cards when they bought their Apple, they would have purchased them at the same time.

GSRAM and GSRAM Plus are available now, allowing up to 8 MEG of memory expansion. That's 8 times the memory capacity of Apple's card and just look at the benefits that only GSRAM and GSRAM Plus have over Apple's card:

- Lower cost
- Has 6 RAM banks (Apple's card has 4)
- Has memory expansion port
- Has ROM expansion port
- No configuration blocks to set
- No soldered in RAM chips
- Expandable to 8 MEG
- Expands AppleWorks internal limits
- Built-in Hi-Res self-diagnostic software
- 5 year hassle free warranty (Apple has a 90 day warranty)
- Made in USA

GSRAM for More AppleWorks Power

Only GSRAM and GSRAM Plus eliminates AppleWorks internal memory limits, increasing the maximum number of records available from 6,000 to over 25,000 and only GSRAM and GSRAM Plus increases the number of lines permitted in the word processing mode from 6,000 to over 15,000. And only GSRAM and GSRAM Plus offers a built-in printer buffer so you can continue using AppleWorks while your printer is printing. GSRAM and GSRAM Plus even expand the number of lines in the clipboard from 255 to 2047 and will auto segment large files so they can be saved on two or more disks. You can even have Pinpoint or Macroworks and your favorite spelling checker in RAM for instant response. GSRAM and GSRAM Plus will even display the time and date right on the AppleWorks screen. Nothing comes close to enhancing AppleWorks so much.

Turn Your IIGS into a Giant

Simply plug GSRAM into the IIGS memory expansion slot and you've got up to *8 megabytes* of RAM at your fingertips—all of it instantly and automatically recognized by the IIGS. GSRAM is compatible with all IIGS software, including AppleWorks, as well as BASIC®, ProDos, DOS 3.3, PASCAL®, "C" and CP/M®.

Grow by Bytes or Megabytes

We offer GSRAM in two configurations so you can increase your memory 256K at a time (GSRAM) or a megabyte at a time (GSRAM Plus). Both offer full compatibility, lower cost than other boards, and easy expandability. And both are extremely low in power consumption. A fully expanded GSRAM operates at only 375 ma, and GSRAM Plus at only 270 ma (even with 6 megabytes on board!)

GSRAM—for Normal Memory Requirements

GSRAM is available with 256K, 512K, 1 MEG or 1.5 MEG of memory already on board. If you don't need the full 1.5 MEG now, you can choose a GSRAM with less memory and expand it up to 1.5 MEG in the future—or upgrade to GSRAM Plus for a small charge.

"I recommend Applied Engineering products wholeheartedly."

Steve Wozniak, the creator of Apple Computer

With an optional piggyback card, you can expand GSRAM even higher than 1.5 MEG! (Other cards are only expandable to 1 MEG.)

GSRAM Plus—for Growing by Leaps and Bounds

GSRAM Plus is the first Apple memory card to use 1 MEG RAM chips on the main board. It's available with 1 to 6 MEG on board. If you don't need the whole 6 MEG now, you can buy a GSRAM Plus with less memory and easily expand it in the future.

GSRAM Plus can be expanded up to 8 MEG with an optional piggyback card.

Easy Expansion

Both GSRAM and GSRAM Plus use standard RAM chips that are readily available and just plug right in. So unlike other cards, you'll find expanding your GSRAM or GSRAM Plus easy, convenient and very economical. And with our optional ROM expansion module you can even increase the IIGS's ROM space and all in just one slot.

Quality and Support of the Industry Leader

Applied Engineering is the largest supplier of Apple peripherals in the world. We *invented* the first large RAM cards for the Apple. With a 5-year "no-hassle" warranty and outstanding technical support, you can be sure GSRAM and GSRAM Plus will deliver the performance you're looking for—or return them within 15 days for a full refund.

GSRAM with 256K	**$169**
GSRAM with 512K	**$219**
GSRAM with 1 MEG	**$299**
GSRAM with 1.5 MEG	**$379**
GSRAM with 2-8 MEG	**CALL**
GSRAM Plus with 1-8 MEG	**CALL**

Order today!

See your dealer or call Applied Engineering today, 9 a.m. to 11 p.m. 7 days. Or send check or money order to Applied Engineering. MasterCard, VISA and C.O.D. welcome. Texas residents add 5⅛% sales tax. Add $10.00 outside U.S.A.

APPLIED ENGINEERING™

The Apple enhancement experts.

(214) 241-6060

P.O. Box 798, Carrollton, TX 75006

CP/M is a registered trademark of Digital Research, Inc.

the string to be printed, relative to the base address QTS. If you read AAL a lot, you have surely seen this little gem a lot.

After assembling the program with the S-C Macro Assembler, type MGO MY.Y to see the state of things when the program is called that way. Then type dollar sign, control-O, control-Y, and the RETURN key, to see the state of things when the ctrl-Y command is used. Or, go to the monitor with the MNTR command, and then just type a control-Y and RETURN. You might also try it with various configurations of hex values before the control-Y.

```
                    1000 *SAVE TEST.CTRL.Y
                    1010 *--------------------------------
                         1020         .OP 65816
                         1030 *--------------------------------
FDED-                    1040 MON.COUT    .EQ $FDED
FDDA-                    1050 MON.PRBYTE  .EQ $FDDA
FD8E-                    1060 MON.CROUT   .EQ $FD8E
                         1070 *--------------------------------
                         1080 T
000800- A9 13            1090         LDA #MY.Y     INSTALL CTRL-Y VECTOR
000802- 8F A1 00 E1      1100         STA $E100A1
000806- A9 08            1110         LDA /MY.Y
000808- 8F A2 00 E1      1120         STA $E100A2
00080C- A9 00            1130         LDA #0
00080E- 8F A3 00 E1      1140         STA $E100A3
000812- 60               1150         RTS
                         1160 *--------------------------------
                         1170 MY.Y
000813- 08               1180         PHP           SAVE MX BITS
000814- 38               1190         SEC
000815- FB               1200         XCE
000816- 08               1210         PHP           SAVE E-BIT
000817- 48               1220         PHA
000818- DA               1230         PHX
000819- 5A               1240         PHY
                         1250 *---Display registers------------
00081A- A0 00            1260         LDY #Q.AEQ
00081C- A3 03            1270         LDA 3,S
00081E- 20 5E 08         1280         JSR DISPLAY.REG
000821- A0 03            1290         LDY #Q.XEQ
000823- A3 02            1300         LDA 2,S
000825- 20 5E 08         1310         JSR DISPLAY.REG
000828- A0 08            1320         LDY #Q.YEQ
00082A- A3 01            1330         LDA 1,S
00082C- 20 5E 08         1340         JSR DISPLAY.REG
00082F- 20 8E FD         1350         JSR MON.CROUT
                         1360 *---Display mode & mx bits-------
000832- A3 04            1370         LDA 4,S       E-BIT
000834- 29 01            1380         AND #1
000836- D0 1A            1390         BNE .1        ...emulation mode
                         1400 *---Native mode------------------
000838- A0 0D            1410         LDY #Q.NATIVE
00083A- 20 73 08         1420         JSR QTO
00083D- A3 05            1430         LDA 5,S       MX BITS
00083F- 29 20            1440         AND #$20      ISOLATE M BIT
000841- 20 66 08         1450         JSR PRINT.0.OR.1
000844- A0 03            1460         LDY #Q.XEQ
000846- 20 73 08         1470         JSR QTO
000849- A3 05            1480         LDA 5,S       MX BITS
00084B- 29 10            1490         AND #$10      ISOLATE X BIT
00084D- 20 66 08         1500         JSR PRINT.0.OR.1
000850- 80 05            1510         BRA .2
                         1520 *---Emulation mode---------------
000852- A0 1E            1530 .1      LDY #Q.EMUL
000854- 20 73 08         1540         JSR QTO
                         1550 *---Return to caller-------------
000857- 7A               1560 .2      PLY
000858- FA               1570         PLX
000859- 68               1580         PLA
00085A- 28               1590         PLP           RESTORE E-BIT
00085B- FB               1600         XCE
00085C- 28               1610         PLP           RESTORE MX BITS
00085D- 60               1620         RTS
```

```
                        1630 *-------------------------------
                        1640 DISPLAY.REG
00085E- 48              1650         PHA
00085F- 20 73 08        1660         JSR QTO
000862- 68              1670         PLA
000863- 4C DA FD        1680         JMP MON.PRBYTE
                        1690 *-------------------------------
                        1700 PRINT.0.OR.1
000866- F0 02           1710         BEQ .1
000868- A9 01           1720         LDA #1
00086A- 09 B0           1730 .1      ORA #"0"      CHANGE TO ASCII "0" OR "1"
00086C- 4C ED FD        1740         JMP MON.COUT
                        1750 *-------------------------------
00086F- 20 ED FD        1760 QTO1    JSR MON.COUT
000872- C8              1770         INY
000873- B9 79 08        1780 QTO     LDA QTS,Y
000876- D0 F7           1790         BNE QTO1
000878- 60              1800         RTS
                        1810 *-------------------------------
                        1820 QTS
00-                     1830 Q.AEQ       .EQ *-QTS
000879- C1 BD           1840             .AS -/A=/
00087B- 00              1850             .HS 00
03-                     1860 Q.XEQ       .EQ *-QTS
00087C- AC A0 D8 BD     1870             .AS -/, X=/
000880- 00              1880             .HS 00
08-                     1890 Q.YEQ       .EQ *-QTS
000881- AC A0 D9 BD     1900             .AS -/, Y=/
000885- 00              1910             .HS 00
0D-                     1920 Q.NATIVE    .EQ *-QTS
000886- CE C1 D4 C9
00088A- D6 C5 A0 CD
00088E- CF C4 C5 BA
000892- A0 A0 CD BD     1930             .AS -/NATIVE MODE:  M=/
000896- 00              1940             .HS 00
1E-                     1950 Q.EMUL      .EQ *-QTS
000897- C5 CD D5 CC
00089B- C1 D4 C9 CF
00089F- CE A0 CD CF
0008A3- C4 C5           1960             .AS -/EMULATION MODE/
0008A5- 00              1970             .HS 00
                        1980 *-------------------------------
```


Navigate your way through ProDOS™ with MouseFiler™

The next generation desktop manager for the Apple® II.

MouseFiler creates a Macintosh™ style desktop for managing disks, directories, programs, and files, and makes the ProDOS file structure easy-to-use and understand. No need to type pathnames. Look at these features!

- Quick access to all files and directories.
- Select and run programs.
- Copy (and paste), rename, compare, delete files and directories.
- Flexible file selection.
- Sort directories on the desktop.
- Work with or without a mouse.
- View the contents of files in ASCII or Hex.
- Copy , compare, rename, format disks.
- Diagnose disks for bad blocks.
- Show space used and free on disks.
- And much more!

Order by sending a check for $39.95 (MA add $2 tax) to:

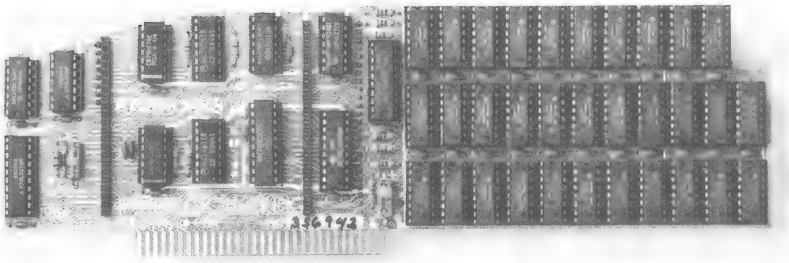
Harbor Software, Inc.
P.O. Box 872
Acton, MA 01720

MouseFiler is compatible with the Apple IIc and Apple IIe (128K enhanced) .

For more information call 1-617-263-1870.

RamWorks® III

Patented Performance from the Recognized Leader



With battery backed RAM port, RGB port, increased memory capacity, full software compatibility and more compact design, RamWorks III is a generation ahead.

RamWorks III is the newest 3rd generation RAM card for the Apple IIe. It incorporates all of the technology and improvements that years of experience and over a hundred thousand sales have given us. By selling more memory cards than anyone else and listening to our customers, we were able to design a memory card that has the ultimate in performance, quality, compatibility and ease of use. A design so advanced it's patented. We call it RamWorks III, you'll call it awesome!

The AppleWorks Amplifier.

While RamWorks III is recognized by all memory intensive programs, NO other expansion card comes close to offering the multitude of enhancements to AppleWorks that RamWorks III does. Naturally, you'd expect RamWorks III to expand the available desktop, after all Applied Engineering was a year ahead of everyone else *including Apple* in offering more than 55K, and we still provide the largest AppleWorks desktops available. But a larger desktop is just part of the story. Look at all the AppleWorks enhancements that even Apple's own card does not provide and *only* RamWorks III does. With a 256K or larger RamWorks III, *all* of AppleWorks (including printer routines) will automatically load itself into RAM dramatically increasing speed by eliminating the time required to access the program disk drive. Switch from word processing to spreadsheet to database at the speed of light with no wear on disk drives.

Only RamWorks eliminates AppleWorks' internal memory limits, increasing the maximum number of records available from 1,350 to over 25,000. *Only* RamWorks increases the number of lines permitted in the word processing mode from 2,250 to over 15,000. And *only* RamWorks offers a built-in printer buffer, so you won't have to wait for your printer to stop before returning to AppleWorks. RamWorks even expands the clipboard. And auto segments large files so they can be saved on two or more disks. You can even have Pinpoint or MacroWorks and your favorite spelling checker in RAM for instant response.

RamWorks, nothing comes close to enhancing AppleWorks so much.

The Most Friendly, Most Compatible Card Available.

Using RamWorks III couldn't be easier because it's compatible with more off-the-shelf software than any other RAM card. Popular programs like AppleWorks, Pinpoint, Catalyst, MouseDesk, Howard-Soft, FlashCalc, Pro-Filer, Managing Your Money, SuperCalc 3a, and MagiCalc to name a few (and *all* hardware add on's like ProFile and Sider hard disks). RamWorks is even compatible with software written for Apple cards. But unlike other cards, RamWorks plugs into the IIe auxiliary slot providing our super sharp 80 column text (U.S. Patent #4601081) in a completely integrated system while leaving expansion slots 1 through 7 available for other peripheral cards.

RamWorks III is compatible with all Apple IIe's, enhanced, unenhanced, American or European versions.

Highest Memory Expansion.

Applied Engineering has always offered the largest memory for the IIe and RamWorks III continues that tradition by expanding to 1 full MEG on the main card using standard RAMs, more than most will ever need (1 meg is about 500 pages of text)...but if you do ever need more than 1 MEG, RamWorks III has the widest selection of expander cards available. Additional 512K, 2 MEG, or 16 MEG cards just snap directly onto RamWorks III by plugging into the industry's only low profile (no slot 1 interference) fully decoded memory expansion connector. You can also choose non-volatile, power independent expanders allowing permanent storage for up to 20 years.

It Even Corrects Mistakes.

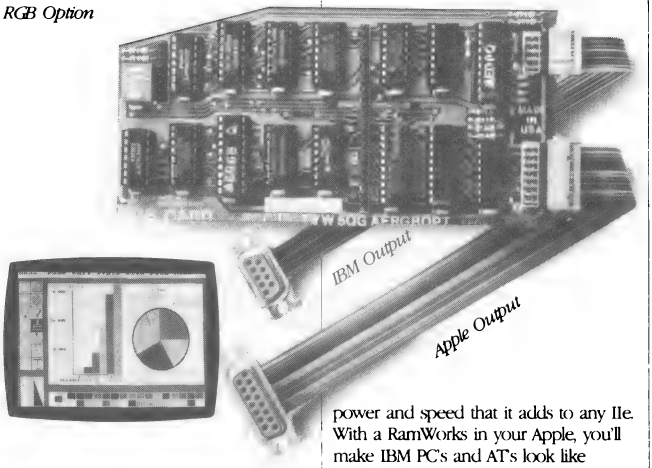
If you've got some other RAM card that's not being recognized by your programs, and you want RamWorks III, you're in luck. Because all you have to do is plug the memory chips from your current card into the expansion sockets on RamWorks to recapture most of your investment!

The Ultimate in RGB Color.

RGB color is an option on RamWorks and with good reason. Some others combine RGB color output with their memory cards, but that's unfair for those who don't need RGB *and* for those that do. Because if you don't need RGB

Applied Engineering doesn't make you buy it, but if you want RGB output you're in for a nice surprise because the RamWorks RGB option offers better color graphics plus a more readable 80 column text (that blows away any composite color monitor). For only $129 it can be added to RamWorks giving you a razor sharp, vivid brilliance that most claim is the best they have ever seen. You'll also appreciate the multiple text colors (others only have green) that come standard. But the RamWorks RGB option is more than just the ultimate in color output because unlike others, it's fully compatible with all the Apple standards for RGB output control, making it more compatible with off-the-shelf software. With its FCC certified design, you can use almost any RGB monitor because only the new RamWorks RGB option provides both Apple standard and IBM standard RGB outputs (cables included). The RGB option plugs into the back of RamWorks with no slot 1 interference and remember you can order the RGB option with your RamWorks or add it on at a later date.

RGB Option



True 65C816 16 Bit Power.

RamWorks III has a built-in 65C816 CPU port for direct connection to our optional 65C816 card. The only one capable of linearly addressing more than 1 meg of memory for power applications like running the Lotus 1-2-3™ compatible program, VIP Professional. Our 65C816 card does not use another slot but replaces the 65C02 yet maintains full 8 bit compatibility.

Endorsed by the Experts.

A+ magazine said "Applied Engineering's RamWorks is a boon to those who must use large files with AppleWorks...I like the product so much that I am buying one for my own system." inCider magazine said "RamWorks is the most powerful auxiliary slot memory card available for your IIe, and I rate it four stars...For my money, Applied Engineering's RamWorks is king of the hill."

"I wanted a memory card for my Apple that was fast, easy to use, and very compatible; so I bought RamWorks."

Steve Wozniak, the creator of Apple Computer

Apple experts everywhere are impressed by RamWorks's expandability, versatility, ease of use, and the sheer power and speed that it adds to any IIe. With a RamWorks in your Apple, you'll make IBM PC's and AT's look like slowpokes.

Quality and Support of the Industry Leader

RamWorks III is from Applied Engineering, the largest, most well supported manufacturer of Apple peripherals and the inventor of large RAM cards for the Apple. With our 5 year no hassle warranty and outstanding technical support, you're assured of the most trouble free product you can buy.

It's Got It All

- 15 day money back guarantee
- 5 year hassle free warranty insures coverage no matter where you purchase
- Built-in super sharp 80 column display, (U.S. Patent #4601081)
- Expandable to 1 MEG on main card
- Expandable to 16 meg with expander cards, with NO slot 1 interference
- Can use 64K or 256K RAMs
- Powerful linear addressing 16 bit coprocessor port
- Automatic AppleWorks expansion up to 3017K desktop
- Accelerates AppleWorks
- Built-in AppleWorks printer buffer
- The only large RAM card that's 100% compatible with all IIe software
- RamDrive™ the ultimate disk emulation software included free
- Memory is easily partitioned allowing many programs in memory at once
- Compatible, RGB option featuring ultra high resolution color graphics and multiple text colors, with cables for both Apple and IBM type monitors
- Built-in self diagnostics software
- Lowest power consumption (U.S. Patent #4601081)
- Takes only one slot (auxiliary) even when fully expanded
- Socketed and user upgradeable
- Software industry standard
- Advanced Computer Aided Design
- Used by Apple Computer, Steve Wozniak and virtually all software companies
- Displays date and time on the AppleWorks screen with any PRO-DOS compatible clock
- Much, much more!

RamWorks III with 64K	**$179**
RamWorks III with 256K	**$219**
RamWorks III with 512K	**$269**
RamWorks III with 1 MEG	**$369**
RamWorks III with 1.5 MEG	**$539**
RamWorks III with 2 to 16 MEG	**CALL**
65C816 16 Bit Card	**$159**
RGB Option	**$129**
Optional Software:	
Pinpoint with RAM Enhancement Software	**$79**
VIP Professional	**$219**

RamWorks III. The industry standard for memory expansion of the Apple IIe.

ORDER YOUR RamWorks III TODAY. 9 a.m. to 11 p.m. 7 days, or send check or money order to Applied Engineering.

MasterCard, Visa and C.O.D. welcome. Texas residents add 5⅛% sales tax. Add $10.00 if outside U.S.A.

Applied Engineering™
P.O. Box 798, Carrollton, TX 75006
(214) 241-6060

How to Use the USR Command.....Bob Sander-Cederlof
in the DOS 3.3 $D000 Version
of the S-C Macro Assembler

If you are running the DOS 3.3 version of my assembler in a 64K or larger Apple, and if you are not using the Laumer Research Full Screen Editor, then you have a 4K bank of RAM available for adding features. It is a little tricky to use, but I am going to show you a secret which will make it easier.

Version 2.0 of the S-C Macro Assembler has several built-in handles for user extensions: USR, PRT, Esc-U, the "." command, and the .US directive. The program below illustrates how to vector the USR directive into the alternate bank at $D000, and how to get back.

Lines 1160-1480 are a program to install your USR code. The first section copies your code into the $D000 bank, and the second section patches the USR vector in the S-C Macro Assembler. I put the origin at $2000 for the installation code, because that is a free area (big enough to hold all the code which will fit at $D000) right after loading the S-C Macro Assembler. You can either manually install it by typing "BRUN B.USR.CODE", or add that line to the EXEC file used to load the S-C Macro Assembler.

The only patch made in the S-C Macro code is to replace the JMP $D028 which is at $D006 with a LDA $C08B instruction. When you type the USR command S-C will jump to $D006, execute the LDA $C08B, and suddenly find itself at $D009 in the alternate bank. That is where your USR program should begin, as you can see in my program below at lines 1570-1620. I made my first instruction in the USR code be another LDA $C08B, which write-enables the RAM. If my code used any of the $Dxxx area for variables, I would need it to be write-enabled.

To get back to the S-C code, the USR program jumps to $D000 (see line 1720 and lines 1510-1530). The instruction there is a LDA $C083, which switches back to S-C's $D000 bank and continues execution at $D003, which is one of S-C's soft entry points.

You could use the six bytes from $D003 through $D008 for variables, if you wish.

You can also link into the PRT command in this same manner, putting the LDA $C08B instruction at $D009 in the S-C bank, and gaining control at $D00C in the alternate bank. If you want both USR and PRT, you can do that too. Just make the instruction at $D009 in the USR-PRT bank a JMP USR, and start the PRT code at $D00C. The "." command can be set up the same way. The .US directive and the Esc-U keyboard function would take a little more setup, because they do not return to the soft re-entry of S-C Macro.

My code at lines 1630-1700 is purely a demonstration, to prove that the USR command really works. Use your imagination, create some really useful functions, and replace my demo with

them.

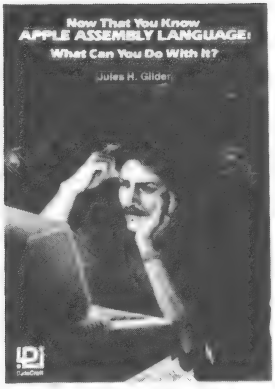
Phil Goetz has sent me the source code for a Symbolic Step/Trace/Disassembly Utility that would fit into this USR patch very nicely. In fact, he had it set up in a similar manner, and his work led me to the discovery of this technique. If all goes well, we should be able to publish his program soon.

Meanwhile, see what you can do with this new tool.

```
                    1000 *SAVE S.USER DEMO
                    1010 *--------------------------------
                    1020 *   Demonstration of linking the USR command
                    1030 *      in the DOS 3.3 version of S-C Macro
                    1040 *      Assembler (loaded at $D000), to a
                    1050 *      program located in the alternate bank
                    1060 *      at $D000.
                    1070 *--------------------------------
                    1080         .OR $2000
                    1090         .TF B.USR.CODE
                    1100 *--------------------------------
00-                 1110 SRCP    .EQ $00,01
02-                 1120 DEST    .EQ $02,03
                    1130 *--------------------------------
D006-               1140 USR.VECTOR .EQ $D006 thru $D008
                    1150 *--------------------------------
                    1160 INSTALLER
2000- AD 8B C0      1170         LDA $C08B     Select alternate $D000 bank
2003- AD 8B C0      1180         LDA $C08B     and write-enable it
                    1190 *---Set up pointers for move-----
2006- A9 3E         1200         LDA #USR.CODE
2008- 85 00         1210         STA SRCP
200A- A9 20         1220         LDA /USR.CODE
200C- 85 01         1230         STA SRCP+1
200E- A9 D0         1240         LDA #$D0
2010- 85 03         1250         STA DEST+1
2012- A0 00         1260         LDY #0
2014- 84 02         1270         STY DEST
2016- A2 01         1280         LDX /USR.CODE.LEN+255
                    1290 *---Move code into destination---
2018- B1 00         1300 .1      LDA (SRCP),Y
201A- 91 02         1310         STA (DEST),Y
201C- C8            1320         INY
201D- D0 F9         1330         BNE .1
201F- E6 01         1340         INC SRCP+1
2021- E6 03         1350         INC DEST+1
2023- CA            1360         DEX
2024- D0 F2         1370         BNE .1
                    1380 *---Install User Vector----------
2026- AD 83 C0      1390         LDA $C083     Select S-C $D000 bank
2029- AD 83 C0      1400         LDA $C083     and write-enable it
202C- A9 AD         1410         LDA #$AD      "LDA $C08B"
202E- 8D 06 D0      1420         STA USR.VECTOR
2031- A9 8B         1430         LDA #$C08B
2033- 8D 07 D0      1440         STA USR.VECTOR+1
2036- A9 C0         1450         LDA /$C08B
2038- 8D 08 D0      1460         STA USR.VECTOR+2
                    1470 *---Return to S-C Macro----------
203B- 4C 03 D0      1480         JMP $D003
                    1490 *--------------------------------
                    1500 USR.CODE   .PH $D000
                    1510 RETURN.TO.SC.MACRO
D000- AD 83 C0      1520         LDA $C083     Next instruction in other bank
                    1530 *                     at S-C Macro "SOFT" entry.
                    1540 *--------------------------------
D003-               1550              .BS 6    These six bytes could be used
                    1560 *                     for variables if you wish.
                    1570 *--------------------------------
                    1580 *   USR.ENTRY must be at $D009, because the USR
                    1590 *   command in S-C is a bank switch at $D006-8.
                    1600 *--------------------------------
```

BECOME AN ASSEMBLY LANGUAGE PROGRAMMING WHIZ

You've spent a lot of time learning Apple assembly language and finally know the difference between BEQ and BCS. Now it's time to put your new-found knowledge to work. Time to throw away your Applesoft programming manual and write programs that make your Apple work like a super-charged, super-fast computer. Time to graduate from the Applesoft BASIC used by beginners, to the 6502 assembly language used by professionals.



To help make this transition, you need an experienced programmer to guide you. You need to develop a library of subroutines that make programming in assembly language as easy as programming in BASIC. You need to learn all the tricks that take experienced assembly language programmers years to acquire. Most important of all, you need the book, *"Now that You Know Apple Assembly Language: What Can You Do With It?"* because it contains all this information *and more.*

It shows you how, step-by-step

"Now That You Know Apple Assembly Language: What Can You Do With It?" will take you step-by-step through the assembly language programming experience. You'll delve into the mysteries of the 6502 stack and learn how to use it to increase the power and versatility of your programs. You'll also learn how to use the Apple's built-in routines to minimize the amount of coding you must do.

Control the output and the input

Frequently it's desirable to gain total control of the computer's output. This book shows you how to *steal control away from the Apple's normal output routines and redirect it to your own program.* Thus if you wanted, you could see the normally invisible control characters, display text on your screen as black on white instead of the normal white on black, format text sent to a printer into pages and much more.

Expand the power of your Apple by *stealing control away from the normal input routines.* Do things like adding a screen print capability, or *convert part of the normal keyboard into a numeric keypad.* It's even possible to *produce self-modifying programs* by EXECing in commands from RAM instead of from the disk drive. Think about the possibilities that offers for protecting your programs. When you want to go back to Applesoft programming, *you'll be able to do it faster with the aid of Applesoft Shorthand,* an assembly language program that types in one or more Applesoft commands at the press of a key, or use another program in the book to *automatically count the number of lines in your Applesoft program.*

With this book you'll also learn about *generating tones and how to figure out the frequency, producing sound effects, teaching your Apple to send Morse code, restoring accidentally erased Applesoft programs, adding new commands to Applesoft and running two Applesoft programs in memory together,* to name a few.

Everything is explained

Unlike other books that merely consist of a collection of programs, this one explains what's happening, where and why. You get detailed descriptions of how the programs work and detailed program listings with virtually every line of code explained. Nothing is left to chance or misinterpretation.

Order now, get 2 FREE gifts

The book costs only $19.95 plus $2 for shipping and handling. Order now and you'll also get a *FREE Programmer's Number Conversion System* that makes it easy to convert between binary, hexadecimal and decimal numbers. No calculators are required. You'll convert numbers almost instantly and wonder how you ever got along without it.

As an extra bonus for prompt ordering, you'll receive a *FREE coupon worth $5 off* the price of a disk with all the assembled programs on it or a disk that contains the source code. These disks normally sell for $15 each. We're offering these FREE gifts for a limited time only, so hurry! *Order today!*

Money-back guarantee*

We're so confident that you'll find this book invaluable and want it in your library, that we're offering a 10-day, no-questions-asked, money-back guarantee. Order the book. Read it and try the programs for ten days. At the end of ten days if you don't think it's worth every penny you paid for it, just send it back in resalable condition and we'll refund your money immediately, no questions asked.

Redlig Systems, Inc., Dept. A 978
2068—79th St., Brooklyn, NY 11214

Please rush me copies of **"Now That You Know Apple Assembly Language: What Can You Do With It?"** at $19.95 each plus $2 shipping and handling. I understand that if I am not delighted with the book I may return it within 10 days for a prompt and courteous refund. In any case, the Programmer's Number Conversion System and $5 coupon are mine to keep.

[ ] Enclosed is my check for $ _____

Please charge my credit card:
[ ] American Express [ ] MasterCard [ ] Visa

Card No. _____ Exp. _____

Signature _____

Name _____

Address _____

City _____ State _____ Zip _____

*NOTE: Shipping and handling fees are not refundable.

```
                     1610 USR.ENTRY
D009- AD 8B C0       1620         LDA $C08B    Finish write-enabling this bank
                     1630 *---User program here-----------
D00C- A0 00          1640         LDY #0       You substitute a REAL program here!
D00E- B9 1F D0       1650 .1      LDA MESSAGE,Y
D011- F0 06          1660         BEQ .2
D013- 20 ED FD       1670         JSR $FDED    PRINT A CHARACTER
D016- C8             1680         INY
D017- D0 F5          1690         BNE .1       ...ALWAYS
D019- 20 E2 FB       1700 .2      JSR $FBE2    RING THE BELL JUST FOR FUN
                     1710 *---Return to S-C Macro----------
D01C- 4C 00 D0       1720         JMP RETURN.TO.SC.MACRO
                     1730 *-------------------------------
                     1740 MESSAGE
D01F- 8D             1750         .HS 8D
D020- D4 E8 E9
D023- F3 A0 E9
D026- F3 A0 EA
D029- F5 F3 F4
D02C- A0 E1 A0
D02F- E4 E5 ED
D032- EF EE F3
D035- F4 F2 E1
D038- F4 E9 EF
D03B- EE             1760         .AS -/This is just a demonstration/
D03C- 8D             1770         .HS 8D
D03D- EF E6 A0
D040- F4 E8 E5
D043- A0 D5 D3
D046- D2 A0 E3
D049- EF ED ED
D04C- E1 EE E4
D04F- AE             1780         .AS -/of the USR command./
D050- 8D 8D 00       1790         .HS 8D.8D.00
                     1800 *-------------------------------
                     1810         .EP
                     1820 *-------------------------------
53-                  1830 USR.CODE.LEN .EQ *-USR.CODE
                     1840 *-------------------------------
```


We Make Measurement And Control Easy!

12 BIT, 16 CHANNEL, PROGRAMMABLE GAIN A/D

- All new 1984 design incorporates the latest in state-of-art I.C. technologies.
- Complete 12 bit A/D converter, with an accuracy of 0.02%!
- 16 single ended channels (single ended means that your signals are measured against the Apple's GND.) or 8 differential channels. Most all the signals you will measure are single ended.
- 9 software programmable full scale ranges, any of the 16 channels can have any range at any time. Under program control, you can select any of the following ranges: ±10 volts, ±5V, ±2.5V, ±1.0V, ±500MV, ±250MV, ±100MV, ±50MV, or ±25MV.
- Very fast conversion (25 micro seconds).
- Analog input resistance greater than 1,000,000 ohms.
- Laser-trimmed scaling resistors.
- Low power consumption through the use of CMOS devices.
- The user connector has +12 and -12 volts on it so you can power your sensors.
- Only elementary programming is required to use the A/D.
- The entire system is on one standard size plug in card that fits neatly inside the Apple.
- System includes sample programs on disk.

PRICE $319

A few applications may include the monitoring of ● flow ● temperature ● humidity ● wind speed ● wind direction ● light intensity ● pressure ● RPM ● soil moisture and many more.

A/D & D/A

A/D & D/A Features:

- Single PC card
- 8 channels A/D
- 8 channels D/A
- Superfast conversion time
- Very easy programming
- Many analog ranges
- Manual contains sample applications

A/D SPECIFICATIONS

- 0.3% accuracy
- On-board memory
- Fast conversion (.078 MS per channel)
- A/D process totally transparent to Apple (looks like memory)
- User programmable input ranges are 0 to 10 volts, 0 to 5, −5 to +5, −2.5 to +2.5, −5 to 0, −10 to 0.

The A/D process takes place on a continuous, channel sequencing basis. Data is automatically transferred to its proper location in the on-board RAM. No A/D converter could be easier to use.

D/A SPECIFICATIONS

- 0.3% accuracy
- On-board memory
- On-board output buffer amps can drive 5 MA
- D/A process is totally transparent to the Apple (just poke the data)
- Fast conversion (.003 MS per channel)
- User programmable output ranges are 0 to 5 volts and 0 to 10 volts

The D/A section contains 8 digital to analog converters, with output buffer amplifiers and all interface logic on a single card. On-card latches are provided for each of the eight D/A converters. No D/A converter could be easier to use. The on-board amplifiers are laser-trimmed during manufacture, thereby eliminating any requirement for off-set nulling.

PRICE $199

SIGNAL CONDITIONER

Our 8 channel signal conditioner is designed for use with both our A/D converters. This board incorporates 8 F.E.T. op-amps, which allow almost any gain or offset. For example, an input signal that varies from 2.00 to 2.15 volts or a signal that varies from 0 to 50 mV can easily be converted to 0-10V output for the A/D.

The signal conditioner's outputs are on a high quality 16 pin gold I.C. socket that matches the one on the A/D's so a simple ribbon cable connects the two. The signal conditioner can be powered by your Apple or from an external supply.

FEATURES

- 4.5" square for standard card cage and 4 mounting holes for standard mounting. The signal conditioner does not plug into the Apple, it can be located up to ½ mile away from the A/D.
- 22 pin .156 spacing edge card input connector (extra connectors are easily available i.e. Radio Shack).
- Large bread board area.
- Full detailed schematic included.

PRICE $79

I/O 32

- Provides 4, 8-Bit programmable I/O Ports
- Any of the 4 ports can be programmed as an input or an output port
- All I/O lines are TTL (0-5 volt) compatible
- Your inputs can be anything from high speed logic to simple switches
- Programming is made very easy by powerful on-board firmware
- The I/O 32 is your best choice for any control application

The I/O manual includes many programs for inputs and outputs.

Some applications include:
Burglar alarm, direction sensing, use with relays to turn on lights, sound buzzers, start motors, control tape recorders and printers, use with digital joystick.

PRICE $89

Please see our other full page ad in this magazine for information on Applied Engineering's Timemaster Clock Card and other products for the Apple.
Our boards are far superior to most of the consumer electronics made today. All I.C.'s are in high quality sockets with mil-spec. components used throughout. P.C. boards are glass-epoxy with gold contacts. Made in America to be the best in the world. All products compatible with Apple II and //e.

Applied Engineering's products are fully tested with complete documentation and available for immediate delivery. All products are guaranteed with a no hassle three year warranty.

Texas Residents Add 5% Sales Tax
Add $10.00 If Outside U.S.A.

Send Check or Money Order to:
APPLIED ENGINEERING
P.O. Box 798
Carrollton, TX 75006

Call (214)241-6060
9 a.m. to 11 p.m. 7 days a week
MasterCard, Visa & C.O.D. Welcome
No extra charge for credit cards

Little Bugs in ProDOS /RAM.....Bob Sander-Cederlof

When a ProDOS device driver reads or writes a block, it is supposed to return an error code in the A-register. If there was an error, this number will be nonzero, and the Carry status bit will be set. If there was no error, the A-register is supposed to contain zero, and Carry will be clear.

The other day I was working with a program that was supposed to read and write blocks from the /RAM drive under ProDOS. My program printed the error code returned by the call, and I was getting a non-zero value in the A-register even when there was no error. I just assumed this was a trivial bug in ProDOS, and ignored it. However, I was not using the standard ProDOS /RAM driver, because I had installed the Applied Engineering driver which uses the entire RAMWORKS card for /RAM. Anyway, I just went on about my business.

A few days later the January 1987 issue of Nibble arrived at my doorstep. In a letter to the editor on page 123, from Steven Humpage, there was an explanation of the bug. The part of the device driver that resides in Main RAM is located at $FF00. At $FF47 is a routine named EXIT, which restores some zero-page locations, restores a vector in page 3, and returns. This code is called with either zero in A and Carry clear, or an error code in A and carry set. The routine begins with PHP, PHA and ends with PLP, PLA. WRONG! This swaps the P- and A-registers. Since the C-bit is bit 0 in the P-register, a zero (from the A-register) results in carry clear. However, the P-register contents come back in A, something like $36. Since the only error code the /RAM driver returns is $27, and this has bit 0 set, swapping registers leaves carry set. It also sets the V-bit, which could be unexpected. The error code in A now becomes what the status was, so we get an error code of $35, I believe.

The same bug exists in AE Prodrive. The bug can easily be fixed, by the following sequence. I am showing it as I do it from within the S-C Macro Assembler. If you do it from within BASIC.SYSTEM, you will have to CALL-151 to get into the monitor to install the patch.

```
:UNLOCK PRODOS                  :UNLOCK PRODRIVE
:BLOAD PRODOS,TSYS,A$2000       :BLOAD PRODRIVE
:$2C5F:68 28                    :$245F:68 28
:BSAVE PRODOS,TSYS,A$2000       :BSAVE PRODRIVE
:LOCK PRODOS                    :LOCK PRODRIVE
```

I am assuming ProDOS 1.1.1. The bug has already been fixed in version 1.2.

While we are at it, Humpage also pointed out another bug which affects both ProDOS and Prodrive. The code which ProDOS puts into pages 2 and 3 in the AuxRAM has an error. The CMP #$0D at $34A should be CMP #$0E. As it is it allows writing to block 7, which is then mapped back over the top of page zero and one, clobbering things disastrously. This change write protects that block. You only need to change the PRODOS file, because Prodrive uses it also. Change $2B4B from $0D to $0E.

New Features in Apple //gs Monitor.....Bob Sander-Cederlof

The //gs Monitor (that which you get by typing CALL -151 in Applesoft, not the video screen) has a lot of neat new features. There are a significant number of new commands, and new capablilities to old ones.

The first thing you need to know is that you can now use 24-bit addresses in the monitor. These are entered by typing the bank number, a slash, and the 16-bit address within the bank. The monitor remembers the last bank number entered, so you do not necessarily have to type the bank number. For example, to examine memory in the monitor ROM itself, you could type the following:

```
*FF/F800.FFFF
```

If you try that, you will notice that both hex and ASCII values are shown for each location. If you are in 40-column mode, you get eight bytes per line as in older monitors; however, in 80-column mode you now get sixteen bytes per line. This means you can easily see an entire page in both hex and ASCII on the screen at once.

Another difference in addressing involves accessing so-called "language card" RAM. In older monitors you could flip the soft switches using monitor commands, and then address the language card. Of course, if you flipped it into a mode to read from the language card RAM, you had to first be sure to make a copy of the monitor ROM in RAM. Anyway, this no longer works. Now you cannot really flip those switches, because the //gs monitor will flip them right back before the next command. Now you can address all the RAM from $C000 to $FFFF in any bank directly.

If you have I/O shadowing on, and you almost always do, then addresses in banks $00 and $01 in the range $C000-CFFF are I/O addresses. The I/O addresses are really in banks $E0 and $E1, but shadowing makes them accessible also in these other banks. To access the $Cxxx addresses in RAM in those four banks, you can see them at $Dxxx by changing the value of a monitor variable "L". Type "1=L" to see the $Cxxx region (2nd bank of language card), or "0=L" to see the real $Dxxx region (1st bank of language card).

That reminds me of another difference. Most of the time the //gs monitor does not care whether you type in upper or lower case. However, some commands are case sensitive. Commands which use a register name, such as the "=" command, are case sensitive. Register names MUST be in the proper case. Some registers must be in upper case, and some must be in lower case.

The disassembler in the //gs monitor handles the entire 65816 opcode set. It has all the expected trouble handling immediate mode disassembly, since there is no way the monitor can tell what mode code will be in when it is executed. You have to control the state of the disassembler's m- and x-bits yourself, by entering commands at the appropriate times. You control

SPECIAL ! ! ! EXPANDED RAM/ROM BOARD: $39.00

Similar to our $30 RAM/ROM dev board described below. Except this board has two sockets to hold your choice of 2-2K RAM, 2-2K ROM or even 2-4K ROM for a total of 8K. Mix RAM and ROM too. Although Apple limits access to only 2K at a time, soft switches provide convenient socket selection. Hard switches control defaults.

IMPROVED ! ! !][IN A MAC (ver 2.0): $75.00

Now includes faster graphics, UniDisk support and more! Bi-directional data transfers are a snap! This Apple II emulator runs DOS 3.3/PRODOS (including 6502 machine language routines) on a 512K MAC or MACPLUS. All Apple II features are supported such as HI/LO-RES graphics, 40/80 column text, language card and joystick. Also included: clock, RAM disk, keyboard buffer, on-screen HELP, access to the desk accessories and support for 4 logical disk drives. Includes 2 MAC diskettes (with emulation, communications and utility software, plus DOS 3.3 and PRODOS system masters, including Applesoft and Integer BASIC) and 1 Apple II diskette.

SCREEN.GEN: $35.00

Develop HI-RES screens for the Apple II on a Macintosh. Use MACPAINT (or any other application) on the MAC to create your Apple II screen. Then use SCREEN.GEN to transfer directly from the MAC to an Apple II (with SuperSerial card) or IIc. Includes Apple II diskette with transfer software plus fully commented SOURCE code.

MIDI-MAGIC for Apple //c: $49.00

Compatible with any MIDI equipped music keyboard, synthesizer, organ or piano. Package includes a MIDI-out cable (plugs directly into modem port - no modifications required!) and 6-song demo diskette. Large selection of digitized QRS player-piano music available for 19.00 per diskette (write for catalog). MIDI-MAGIC compatible with Apple II family using Passport MIDI card (or our own input/output card w/drum sync for only $99.00).

FONT DOWNLOADER & EDITOR: $39.00

Turn your printer into a custom typesetter. Downloaded characters remain active while printer is powered. Use with any Word Processor program capable of sending ESC and control codes to printer. Switch back and forth easily between standard and custom fonts. Special functions (like expanded, compressed etc.) supported. Includes HIRES screen editor to create custom fonts and special graphics symbols. For Apple II, II+, //e. Specify printer: Apple Imagewriter,Apple Dot Matrix, C.Itoh 8510A (Prowriter), Epson FX 80/85, or Okidata 92/192.

*** FONT LIBRARY DISKETTE #1: $19.00** contains lots of user-contributed fonts for all printers supported by the Font Downloader & Editor. Specify printer with order.

DISASM 2.2e : $30.00 ($50.00 with SOURCE Code)

Use this intelligent disassembler to investigate the inner workings of Apple II machine language programs. DISASM converts machine code into meaningful, symbolic source compatible with S-C, LISA, ToolKit and other assemblers. Handles data tables, displaced object code & even provides label substitution. Address-based triple cross reference generator included. DISASM is an invaluable machine language learning aid to both novice & expert alike. Don Lancaster says DISASM is "absolutely essential" in his ASSEMBLY COOKBOOK.

The 'PERFORMER' CARD: $39.00 ($59.00 with SOURCE Code)

Converts a 'dumb' parallel printer I/F card into a 'smart' one. Simple command menu. Features include perforation skip, auto page numbering with date & title, large HIRES graphics & text screen dumps. Specify printer: MX-80 with Graftrax-80, MX-100, MX-80/100 with Graftraxplus, NEC 8092A, C.Itoh 8510 (Prowriter), OkiData 82A/83A with Okigraph & OkiData 92/93.

'MIRROR' ROM: $25.00 ($45.00 with SOURCE Code)

Communications ROM plugs directly into Novation's Apple-Cat Modem card. Basic modes: Dumb Terminal, Remote Console & Programmable Modem. Features include: selectable pulse or tone dialing, true dialtone detection, audible ring detect, ring-back, printer buffer, 80 col card & shift key mod support.

RAM/ROM DEVELOPMENT BOARD: $30.00

Plugs into any Apple slot. Holds one user-supplied 2Kx8 memory chip (6116 type RAM for program development or 2716 EPROM to keep your favorite routines on-line). Maps into $Cn00-CnFF and $C800-CFFF.

C-PRINT For The APPLE //c: $69.00

Connect standard parallel printers to an Apple //c serial port. Separate P/S included. Just plug in and print!

Unless otherwise specified, all Apple II diskettes are standard (not copy protected!) 3.3 DOS.

Avoid a $3.00 handling charge by enclosing full payment with order. VISA/MC and COD phone orders OK.

RAK-WARE 41 Ralph Road W. Orange N J 07052 (201) 325-1885

their state by typing any of these four commands:

```
0=m    1=m    0=x    1=x
```

This can be pretty messy and painful, but it is one of those things we have to put up with. Of course, the monitor could be a little smarter, and TRY to follow the XCE, REP, and SEP commands, assuming that normal code will be "nice". Then we could just override when absolutely necessary. However, we have to live with what is there for now.

The disassembler now puts out a status line at the top of each set of 20 lines, which shows the state of the m- and x-bits, and the L-variable mentioned above. This is somewhat of a nuisance when using "LLLLLLL" to print out a long range, but we can live with it too. Another difference is that BRK disassembles with a one-byte value in the operand field. This has always been the correct way to handle BRK, but nobody knew it in the assembler and dis-assembler business until the 65816 came out.

The mini-assembler is also included in the //gs monitor. You get into it by using the "!" command. The prompt changes from "*" to "!". You can leave the mini-assembler and go back to the monitor by typing a RETURN key immediately after the prompt. Operation inside the mini-assembler is just like the old mini-assembler, with a few enhancements. You can assemble to any bank of RAM, specifying the bank before a slash. All of the 65816 opcodes can be assembled. You indicate 16-bit immediate by typing 3- or 4-digit values after a "#", 8-bit immediate by typing 1- or 2-digit values.

You can also enter ASCII and hex values while in the mini-assembler. Type a quotation mark followed by the characters you wish to enter in ASCII mode. Terminate the string with a RETURN. Type a colon to indicate that you are entering hex values. The number of bytes of hex data you store depends on the number of digits you type; you can type from 1 to 8 digits, occupying 1 to 4 bytes. Multi-byte values will be stored with the least significant byte first. Practice with these a while and you will get the hang of it.

Step and Trace commands are included in the //gs monitor, but they are not "plugged in". They point to vectors in bank $E1, and all they do now is print "Step" and "Trace". The TRACE vector is at $E1/0074, and the STEP vector is at $E1/0078. By loading a real Step/Trace program into RAM and hooking into the vectors, you can get them to work. I have seen them work, but I do not know where to put my hands on the working code or how to load it if I had it. Do any of you?

Here are the commands, with brief commentary. (The ^ means "Control-"; "addr" may be in form "bb/aaaa" or "aaaa".)

```
^B              JMP $00E000, initial entry to Applesoft.
^C              JMP $00E003, re-enters Applesoft.
Q               JMP $0003D0, re-enters DOS or ProDOS.
^Y              JMP $E000A0, which normally does JMP $0003F8.

^E              Examine Stored Registers.  Displays the contents
                of A X Y S D P B K M Q L m x e.
^R              Restore Stored Registers to normal monitor mode:
                A=0000 X=0000 Y=0000 S=01FB D=0000 P=00
                B=00 K=00 M=0C Q=80 L=1 m=1 x=1 e=1
value=reg       Assign value to a stored register.  The register
                names are case-sensitive:  A X Y S D P B K M Q
                m x e L F.  ("F" is Filter for entering ASCII,
                and will normally be 7F or FF.)

s^K             Connect input hook to slot s.
s^P             Connect output hook to slot s.

^S              During output, pause-restart.  (There appears
                to also be an entry at the end of the command
                table for ^S, but it is not really accessible
                due to the loop counter used.)
^X              During memory-range display, terminate command.
                (This, like ^S, is not really a command.)

^T              Similar to Applesoft TEXT command, changes
                screen display to text page 1.
I               Switch to Inverse display mode.
N               Switch to Normal display mode.

^6c             Change cursor to new character c.
                Control-6 DELETE restores original cursor.
                Control-6 UNDERLINE is especially nice.

addrG           Pick up stored registers and JSR addr.
                Only works properly in bank 00, because
                assumes user returns with RTS.
addrX           Pick up stored registers and JSL addr.
                Assumes user returns with RTL.
addrR           Pick up stored registers and JMP addr.

addrL           Disassemble 20 lines starting at addr.
L               Disassemble next 20 lines.

addr<addr.addrM   Move the range to the target.
addr<addr.addrV   Compare the range to the target.
zz<addr.addrZ     Fill the range with single byte value zz.

\pattern\<addr.addrP   Search range for pattern, which may
                contain hex values of 1 to 8 digits, hi-ASCII
                strings ("abcd"), lo-ASCII strings ('abcd'),
                and may be up to 236 bytes long!

\#bytesin #bytesout parm1 ... parmN func# tool#\U
                Call Tool (tool#) in Toolbox (func#) with
                parameters (parm1 ... parmN) on the stack.
                All the items between the \\ must be separated
                with spaces. #bytesin is total # of bytes
```

```
                that will be pushed onto the stack before
                calling the tool.  #bytesout is the # of bytes
                the tool will leave on the stack.  Wow!
                I dare you to try it!

addrS or S      Step, only if loaded into RAM
addrT or T      Trace, only if loaded into RAM

hexvalue=       convert hexvalue to decimal
=decvalue       convert decvalue to hexadecimal

=T              display date & time mm/dd/yy hh:mm:ss
=T=mm/dd/yy hh:mm:ss    Set date & time
                (both must be capital T)

addr            Display contents of addr.
addr.addr       Display contents of range.
addr:stuff      Enter information in hex or ASCII.
value+value     add
value-value     subtract
value*value     multiply
value_value     divide
```


EnterSoft

Basic-like macros for the S-C Macro Assembler.

Available for the Apple][,][+, //e, and //c.

Uses whatever DOS your S-C Macro Assembler uses.

Soon to be released for PRODOS.

Source code included with every disk!!!

Cost? A paltry $30.00!!!

To Order - Send Check or Money Order to:

Simulacron I
Baggy Games

Mark Manning
c/o Baggy Games
P. O. Box 591894
Houston, Texas 77259-1894

Apple is a registered Trademark of Apple Computer Inc.

The S-C Macro Assembler is produced by The S-C Software Corporation.

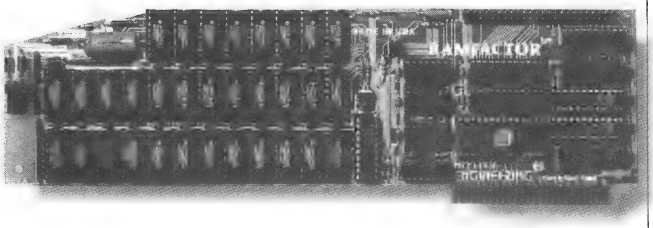
EnterSoft is produced by Mark Manning/Simulacron I/Baggy Games

RamFactor™

The Highest Performance, Speed and Software Compatibility of Any Slot 1-7 Memory Card for the Apple II+, IIe or IIGS.

Now anyone with an Apple II+, Franklin, Laser 128, Apple IIe or IIGS preferring to use slots 1 through 7 can now enjoy the speed and performance that they've been waiting for.

With RamFactor, you'll be able to instantly add another 256K, 512K, or a full 1 MEG on the main board and up to 16 MEG with additional piggyback card. And since virtually all software is automatically compatible with RamFactor, you'll immediately be able to load programs into RamFactor for instantaneous access to information. You'll also be able to store more data for larger word processing documents, bigger data bases, and expanded spreadsheets. And with the battery back-up option you can switch on your Apple and any of your favorite software will be up and running in less than 1 second!



Very Compatible

All the leading software is already compatible with RamFactor. Programs like AppleWorks, Pinpoint, BPI, Managing Your Money, Dollars and Sense, SuperCalc 3A, PFS, MouseWrite, MouseDesk, MouseCalc, Sensible Speller, Applewriter IIe, Business Works, ReportWorks, Catalyst 3.0 and more. And RamFactor is fully ProDos, DOS 3.3, Pascal 1.3 and CP/M compatible. In fact, no other slot 1-7 memory card is more compatible with commercial software.

AppleWorks Power

There are other slot 1-7 cards that give AppleWorks a larger desktop, but that's the end of their story. But RamFactor is the only slot 1-7 card that increases AppleWorks internal memory limits, increasing the maximum number of records in the database and lines permitted in the word processor, and RamFactor is the only standard slot card that will automatically load all of AppleWorks into RAM dramatically increasing speed and eliminating the time required to access the program disk, it will even display the time and date on the AppleWorks screen with any ProDos clock. RamFactor will automatically segment large files so they can be saved on 5¼", 3½", and hard disks. All this performance is available to anyone with an Apple IIe, IIGS *or* II+ with an 80 column card. RamFactor, no other standard slot card comes close to enhancing AppleWorks so much.

True 65C816 16 Bit Power

RamFactor has a built-in 65C816 CPU port for direct connection to our IIe 65C816 card for linearly addressing up to 16 MEG for the most powerful 16 bit applications. (II+ 65C816 card under development.)

Powerful Program Switcher

With RamFactor, you can organize memory into multiple work areas and switch between them. Each work area can contain different programs and even different operating systems. Now you can switch from one program to another or even switch from AppleWorks to DOS 3.3 to CP/M to Pascal to ProDos in under a second. And with our Battery back-up option, you can have permanent storage for up to 20 years.

Quality and Support of the Industry Leader

RamFactor is from Applied Engineering, the largest, most well supported manufacturer of Apple peripherals and the inventor of large RAM cards for the Apple. With our 5 year no hassle warranty and outstanding technical support, you're assured of the most trouble free product you can buy.

Features:

- Up to 16 MEG total memory, 256K to 1 MEG on main board. Up to 16 MEG with additional memory on piggyback card.
- Fully Apple II Memory Expansion compatible
- Compatible with Apple IIe, II+, IIGS, Franklin and Laser 128
- Battery back-up option allows you to turn on your Apple and run your favorite programs in less than 1 second!
- Automatically recognized by ProDos, DOS 3.3, Pascal and standard CP/M
- Built-in RamDrive™ software (a true RAM disk not disk caching)
- Systems are directly bootable from RamFactor if desired
- Built-in linear addressing 16 bit co-processor port
- Built-in self diagnostic software
- Automatic expansion with AppleWorks 1.3 or later
- Allows Apple II+ and IIe to run your AppleWorks without buying additional software
- Accelerates AppleWorks
- Displays time and date on the AppleWorks screen with any ProDos clock
- Fits any I/O slot except slot 3
- Fully socketed and user upgradeable
- Much, much more

RamFactor with 256K	$239
RamFactor with 512K	$289
RamFactor with 1 MEG	$389
RamFactor with 2-16 MEG	CALL
Battery Back-up Option	$179
65C816 16 Bit Card	$159

Order RamFactor today . . . with 15 day money back guarantee and our "no hassle" five year warranty. *See* your dealer *or call* (214) 241-6060, 9 a.m. to 11 p.m., 7 days, *or send* check or money order to Applied Engineering. MasterCard, Visa and C.O.D. welcome. Texas residents add 5⅛% sales tax. Add $10.00 if outside U.S.A.

Applied Engineering™

The Apple enhancement experts.

(214) 241-6060

P.O. Box 798, Carrollton, TX 75006

New ProDOS Version of S-C XREF.....Bob Sander-Cederlof

An upgrade is now available for the ProDOS version of the S-C XREF program. XREF stands for Cross Reference, and if you are working on large programs with the S-C Macro Assembler you really need it. If you already have it, and are developing programs under ProDOS, you will enjoy this new upgrade.

The new version still fits in the same space, loading between $800 and $FFF. (It is actually a little shorter than the previous release.) There are, however, a lot of new features.

First and foremost, I added the ability to automatically handle a source program which uses the .IN or .INB directive for multiple source files. The previous release required you to create a complex EXEC file to control XREF if you wanted to make a complete cross reference for a program that was spread over several files. Sometimes a lot of juggling was required. For example, when I run XREF on the source code of the S-C Macro Assembler (which consists of over two dozen files on two floppies), I had to make two passes: one pass for symbols beginning with A-L, another for symbols starting with M-Z. The EXEC file for each consisted of over 50 lines. Too much!

Anyway, with the new version I can simply LOAD the main assembly control file (the source file containing the .IN directives) and type "-XREF". The new version automatically letters each included file for purposes of identifying where each cross reference came from. The main file, which contains the .IN directives, is identified by not having a letter. In other words, its identifier is a blank. The first 26 included files are lettered A-Z; the next 26 are lettered a-z; the next 10 are numbered 0-9; if you have another, it gets a punctuation symbol (:). Sixty-four files should be enough!

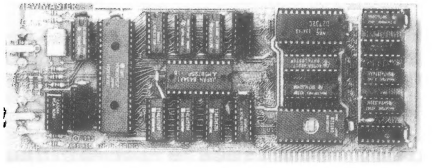
One reason I have so many separate source files for one program is so that I could squeeze out a meaningful cross reference. With this new version the file size is not so important. During operation, when XREF sees an include directive (either .IN or .INB), only 512 bytes are needed for reading the file. It is read in only one block at a time, equivalent to the normal .INB or .INB1. This leaves a lot more room for symbols.

The new version also prints a legend of the included files, showing the assigned letter or number and the file name for each. This makes it a lot easier to read the printed cross reference listing.

I simplified the syntax analysis portion of XREF, so that tables particularizing it to the opcodes of a single microprocessor are no longer necessary. This means that the same XREF program can operate on the source code of any of our assemblers. Not only 6502, 65C02, and 65816 code, but also 6800, 6805, etc. Now if we could just upgrade all the rest of our cross assemblers to run under ProDOS.... The purpose (in the previous version) for opcode tables was to inform XREF of which opcodes had no operands, and which if any had the option to be used both with and without operands. This is no longer

Viewmaster 80

Now run AppleWorks™ on your II+ *and* keep full 80 column compatibility!



	BUILT-IN SOFTSWITCH	SHIFT KEY SUPPORT	LOW POWER DESIGN	80 COLUMN MODE	7 x 9 DOT MATRIX	LIGHT PEN DOWTS	40 COLUMN OVERRIDE	INVERSE CHARACTERS	
VIEWMASTER 80	✓	✓	✓	✓	✓	✓	✓	✓	
SUPRTERM		✓					✓	✓	
WIZARD 80					✓		✓	✓	
VISION 80	✓	✓			✓				
OMNIVISION		✓					✓	✓	
VIEWMAX 80	✓	✓			✓			✓	
SMARTERM	✓	✓				✓	✓		
VIDEOTERM		✓	✓		✓	✓		✓	

One look at the chart will give you some of the reasons there's only one smart choice in 80 column cards for your Apple. But the real secret to Viewmaster 80's success is something even better: Total compatibility.

Each Viewmaster 80 includes our powerful AppleWorks expand software, allowing AppleWorks to run on the II+ with only 64K (or more) memory. (We recommend the RamFactor memory card, but any compatible Apple memory card will work.) The software provides our full range of AppleWorks enhancements, including expanded records, word processor, multi-disk saving, time and date display on screen with any PRO-DOS clock, and more!

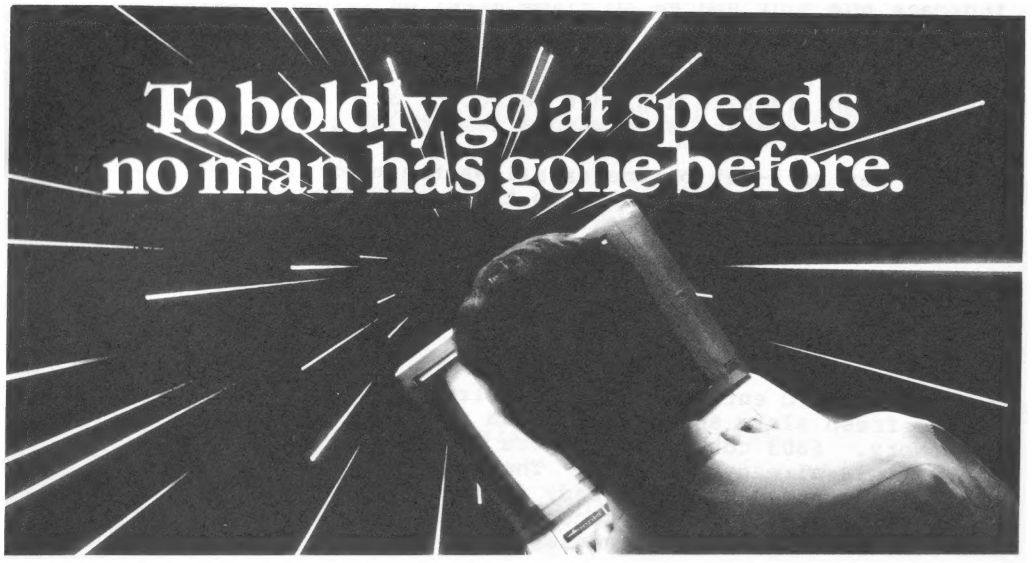
The Viewmaster 80 works with all 80 column applications, including DOS 3.3, PRO-DOS, CP/M, Pascal, WordStar, Format II, Applewriter II, dBase II, Visicalc, Multiplan, and *hundreds* of others—including AppleWorks.

Here are just a few of the powerful features the Viewmaster 80 delivers for a great price ($149):
• 80 Characters by 24 lines • Fully compatible with all Apple languages and software • Highest compatibility with existing 80 column software • Very low power consumption • High speed (18 MHZ) scroll rate • Upper and lower case characters and true descenders, both inverse and normal; all on-screen editing functions are supported • User-definable cursor shape • Compatible with Apple II, II+ and IIe • Five-year warranty

Call today to order or for more information, 9 a.m. to 11 p.m. seven days, or send check or money order to Applied Engineering. MasterCard, VISA and C.O.D. welcome. Texas residents add 5¼% sales tax. Add $10.00 outside U.S.A.

CP/M is a registered trademark of Digital Research, Inc.

Applied Engineering
P.O. Box 798, Carrollton, TX 75006
(214) 241-6060



With the TransWarp™ accelerator, your IIe™ or II+™ is 40% faster than the IIGS!

Computing at warp speed!

It's an experience you shouldn't miss. And with TransWarp, you won't have to. Because TransWarp will run your IIe or II+ software *3.6 times faster* — that's 40% faster than Apple's IIGS!

No more yawning while your Apple™ slowly rearranges text or calculates spreadsheets. With 256K of ultra-fast RAM, TransWarp speeds up *all* Apple software — including Apple Works, and all educational software, graphics and games. And it's compatible with all standard peripheral cards (such as RamWorks and Apple memory cards), hard disks, 3½" UniDisks, 80-column cards, modems, clock cards, mouses and more! You name it, TransWarp accelerates it. There's even a 16 bit upgrade chip available.

"TransWarp is great! I have replaced all my other accelerators with it!"

Steve Wozniak, the creator of Apple Computer

An important difference.

With TransWarp, software runs up to three times faster than with other cards, since the others can't accelerate programs in auxiliary memory. That's why TransWarp is so much faster than the rest. Nearly all of today's more powerful programs run partially or completely in auxiliary memory: programs like Apple Works, Pinpoint, Managing Your Money, SuperCalc 3a, BPI and Pascal, to name a few. Why settle for a card that accelerates only part of the memory?

There's one more important difference. Since TransWarp doesn't use memory caching, you get *consistent* high speed performance.

A cinch to use.

Simply plug TransWarp into any slot in your Apple II, II+ or IIe — including slot 3 in the IIe. Instantly you'll be computing at speeds you only dreamed about before. And should you ever wish to run at normal speed, simply press the ESC key while turning your Apple on.

Since TransWarp is completely transparent, you won't need preboot disks or special software. It's ready to go right out of the package!

Speed = Productivity

Imagine the productivity gains you'll achieve when your programs are running over three times faster. TransWarp is so powerful, your Apple will make IBM PCs™ and even ATs™ look like slowpokes.

- 3.6 MHZ 65C02
- 256K of ultra-fast on board RAM
- Accelerates main *and* auxiliary memory
- Low power consumption for cool operation
- Totally transparent operation with all software
- Plugs into any slot, including slot 3 on the Apple IIe
- Accelerated 16 bit option available
- 5 year warranty

Satisfaction guaranteed!

Give *your* Apple the TransWarp advantage. With our risk-free 15-day money back guarantee, you have nothing to lose but wasted time. Call today!

TransWarp accelerator	$279
16 bit upgrade (may add later)	$89

For fast response:

Call Applied Engineering, 9 a.m. to 11 p.m., 7 days at (214) 241-6060. MasterCard, VISA and C.O.D. welcome. Texas residents add 5⅛% sales tax. Add $10.00 if outside U.S.A. *Or mail* check or money order to Applied Engineering, P.O. Box 798, Carrollton, TX 75006.

Applied Engineering

The Apple enhancement experts.

P.O. Box 798, Carrollton, TX 75006 (214) 241-6060

necessary. Instead, I use a single user-set parameter to indicate how many spaces to allow after an opcode before believing the comment field has been reached. The default for this parameter is two, which works well with the 6502 family. Some other micorprocessors, which have opcode names of varying length, might require a higher value. I also added code to terminate the operand scan when a semi-colon is encountered.

Another new user-set parameter allows the user to choose to ignore the ,X and ,Y indexing in the cross reference listing. I personally found no benefit to seeing those two large entries, since X and Y were not really symbol names. The default position for this parameter avoids listing references to X and Y if they are only referenced and never defined within the program. If you liked it the old way, simply change the parameter setting.

I added a third entry point. The first one, at $800, starts with a fresh slate and does a cross reference for the program in memory. $803 continues a cross reference, without erasing the symbols already gathered. The new one, at $806, lists the cross reference that is already in the symbol table. I find this useful when I need a second copy and do not wish to wait for all the files to be processed over again.

I fixed some problems in the processing of the .TI directive, so that titles now appear properly on the cross reference listing. Of course, if you have more than one .TI directive in your source code, the entire cross reference listing will use the last one.

There are more improvements, but this is enough detail for this article. If you already own the S-C XREF, you can get the upgraded ProDOS version for $10. If you purchased the source code with your XREF, we will include the new source code too. If you do not have S-C XREF yet, the price is $50 including source code, with the DOS version on one side and the new ProDOS version on the other.

If you have been thinking about typing in the sample programs from Beneath Apple ProDOS, but just haven't gotten around to it yet, there is an easy way out. Pete Johnson has done all the hard part for you. He typed them in and got them running, after a lot of head-scratching. If you send him $2 and a blank diskette, he will send you a copy. The source files are in the S-C Macro Assembler format, and he includes the binary object files also. The programs are ZAP, FORMAT, TYPE, FIB, and DUMP. He does not include MAP (which is in Basic) and DUMBTERM. Pete's address is P. O. Box 5, New London, Minnesota 56273.

Apple Assembly Line (ISSN 0889-4302) is published monthly by S-C SOFTWARE CORPORATION, P.O. Box 280300, Dallas, Texas 75228. Phone (214) 324-2050. Subscription rate is $18 per year in the USA, sent Bulk Mail; add $3 for First Class postage in USA, Canada, and Mexico; add $14 postage for other countries. Back issues are $1.80 each (other countries inquire for postage). A subscription to the newsletter and the Monthly Disk containing all source code is $64 per year in the US, Canada and Mexico, and $87 to other countries.

All material herein is copyrighted by S-C SOFTWARE CORPORATION, all rights reserved. (Apple is a registered trademark of Apple Computer, Inc.)